International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 04 April-2026 | Impact Factor: 3.5

Real-Time Self-Evolving Intelligence System with LLM Integration for Multi-Tenant Invoice Anomaly Detection

Aniket Kumar
Computer Science And Engineering, SRM Institute of Science & Technology,
Email ID: [ak7525@srmist.edu.in]
Pawan Kumar
Computer Science And Engineering,SRM Institute of Science & Technology,
Email ID: [pk2806@srmist.edu.in]
Eklavya Singh
Computer Science And Engineering, SRM Institute of Science & Technology,
Email ID :-[es3044@srmist.edu.in]

How to Cite this Article:
Singh, E., Kumar, P. & Kumar, A. (2026). Real-Time Self-Evolving Intelligence System with LLM Integration for Multi-Tenant Invoice Anomaly Detection. International Journal of Creative and Open Research in Engineering and Management, 02(04). https://doi.org/10.55041/ijcope.v2i4.889

https://doi.org/10.55041/ijcope.v2i4.889

ABSTRACT: *This paper presents a self-evolving, schema-adaptive anomaly detection system designed specifically for multi-tenant billing and invoicing environments. Contemporary enterprise billing pipelines suffer from persistent vulnerabilities arising from static rule-based validation frameworks that cannot adapt to the distributional evolution of transactional data over time. The proposed architecture addresses this fundamental limitation through the integration of three synergistic components: automatic schema-aware feature engineering, an incremental Isolation Forest anomaly detection engine, and a Large Language Model (LLM)-driven reasoning and explanation layer. Upon onboarding a new tenant, the system performs unsupervised schema introspection to classify data attributes and derive semantically meaningful feature representations. The anomaly detection model is subsequently trained on tenant-specific historical invoice data and undergoes periodic retraining through a human-feedback-driven incremental learning loop. When anomalous invoices are identified at runtime, the system computes feature-level deviation signatures and submits structured deviation reports to an integrated LLM, which generates contextually grounded explanations and corrective recommendations. Experimental evaluation demonstrates that this hybrid intelligence framework substantially reduces false-positive anomaly rates while improving detection sensitivity across diverse invoice schemas. The feedback-driven retraining mechanism enables sustained accuracy gains over deployment lifetime, and the LLM explanation layer measurably enhances operator trust and response effectiveness. This work constitutes a meaningful contribution toward intelligent, self-adaptive financial validation systems capable of evolving autonomously with changing business patterns and invoice structures.*

KEYWORDS: *Anomaly Detection, Isolation Forest, Large Language Model, Invoice Validation, Schema Adaptation, Incremental Learning, Multi-Tenant Systems, Feature Engineering, Concept Drift, Intelligent Billing Systems, Human-in-the-Loop, Self-Evolving Intelligence*

1. INTRODUCTION

1.1 Background and Motivation

Enterprise financial operations generate enormous volumes of transactional invoice data daily. Across logistics, healthcare, manufacturing, and retail sectors, the accuracy and integrity of invoice processing directly determines cash-flow reliability, regulatory compliance, and audit readiness. Yet the validation mechanisms underlying most contemporary billing platforms remain fundamentally static—they rely on deterministic rule sets that must be manually authored, maintained, and updated as business requirements evolve. Such rule-based approaches cannot detect complex multivariate anomalies that emerge from the interaction of multiple invoice attributes simultaneously, nor can they adapt autonomously to structural changes in invoice schemas arising from business growth, vendor onboarding, or product-line diversification.The consequences of inadequate invoice validation are substantial. Fraudulent billing exploits, unintentional data-entry errors, and computational inconsistencies in charge aggregation collectively account for significant financial losses in enterprise supply chains. According to industry surveys, billing disputes and payment delays attributable to invoice inaccuracies impose measurable operational overhead that erodes profitability margins, particularly in multi-tenant software-as-a-service billing environments where a single platform must simultaneously accommodate the heterogeneous invoice schemas of dozens or hundreds of distinct client organizations.Machine learning-based anomaly detection offers a principled alternative to rule-based validation. Unlike deterministic validators, learned models can characterize the high-dimensional joint distribution of invoice attributes from historical data and identify observations that deviate from learned behavioral norms. However, practical deployment of such systems in multi-tenant invoice contexts introduces distinct challenges that are not adequately addressed by existing generic anomaly detection literature. These challenges include automatic schema discovery and feature engineering without human intervention, graceful adaptation to the distinct statistical properties of each tenant's invoice population, continuous model evolution as invoice patterns shift over time, and the generation of human-interpretable explanations that enable operators to evaluate and act upon detected anomalies effectively.

1.2 Research Problem Statement

The central research problem addressed in this paper is the design and validation of a self-evolving anomaly detection architecture that can operate effectively across multiple tenants with heterogeneous invoice schemas, adapt continuously to shifting invoice distributions, and provide actionable LLM-generated explanations that bridge the gap between statistical anomaly scores and human decision-making. The system must achieve this without requiring schema-specific manual configuration for each tenant, and must improve over time through structured feedback from invoice operators.

1.3 Research Contributions

This work makes the following principal contributions to the field of intelligent financial data validation:

(i) A schema-aware automatic feature engineering pipeline that performs unsupervised column type classification and generates contextually appropriate feature transformations for numeric, categorical, temporal, and textual invoice attributes without manual intervention;

(ii) An adaptive multi-phase anomaly detection engine based on Isolation Forest with incremental retraining capabilities, enabling sustained detection accuracy across distributional shifts and tenant-specific behavioral evolution;

(iii) An LLM-based reasoning layer that transforms feature-level deviation reports into structured, human-readable explanations and corrective recommendations, substantially improving operator interpretability and intervention effectiveness;

(iv) A human-in-the-loop feedback integration mechanism that systematically accumulates operator validation decisions to drive periodic model retraining, enabling the system to self-correct false-positive tendencies and align with evolving business norms;

(v) A shadow model deployment strategy that maintains a parallel candidate model trained on recent feedback-enriched data and promotes it upon demonstrating statistically superior performance over the production model.

The remainder of this paper is organized as follows. Section 2 reviews related work in anomaly detection, schema-adaptive machine learning, and LLM-based data validation. Section 3 presents the overall system architecture. Sections 4 through 6 detail the three core subsystems. Section 7 describes the experimental methodology and results. Section 8 discusses key findings and practical implications, and Section 9 concludes the paper with directions for future research.

2. RELATED WORK

2.1 Anomaly Detection in Financial and Transactional Data

The application of machine learning to financial anomaly detection has a well-established research lineage. Early approaches relied on threshold-based statistical rules and linear discriminant analysis [1]. Subsequent advances introduced ensemble methods and kernel-based techniques, among which Isolation Forest [2] emerged as a particularly effective algorithm for unsupervised anomaly detection owing to its computational efficiency and robustness to high-dimensional data. The Isolation Forest algorithm operates on the premise that anomalous observations, being rare and structurally distinct from the majority, are more rapidly isolated by recursive random partitioning of the feature space. Its average-case time complexity of O(n log n) and linear scaling with dataset size make it well-suited for the real-time invoice processing context targeted in this work.One-Class SVM [3] provides an alternative formulation for anomaly detection when only normal-class examples are available during training, which aligns with the operational reality of invoice validation where labelled anomaly examples are scarce during initial deployment. Autoencoder-based methods [4] have more recently gained traction for detecting anomalies through reconstruction error analysis, and variational autoencoders have been applied to fraud detection with measurable improvements in recall. However, none of these prior approaches address the combined challenges of automatic schema adaptation, multi-tenant isolation, and LLM-mediated explanation that distinguish the system proposed in this work.

2.2 Concept Drift and Adaptive Machine Learning

The theoretical foundations of concept drift detection and adaptation are rigorously established in the work of Yu et al. [5], who proposed a hierarchical hypothesis testing framework (HLFR) for streaming data classification. The HLFR framework employs a two-layer statistical testing architecture in which a Layer-I test based on Linear Four Rates (LFR) monitors changes in confusion matrix statistics, and a Layer-II permutation test validates or rejects detected drift points to reduce false alarms. This work demonstrated that hierarchical drift detection frameworks can achieve substantially lower false-positive rates compared to single-layer approaches such as Drift Detection Method (DDM) [6] and Early Drift Detection Method (EDDM) [7], while preserving recall performance. The adaptive SVM training strategy proposed alongside HLFR, which regularizes the distance between new and previous classifier parameters, provides a formal mechanism for knowledge transfer across concept boundaries.

The present work draws conceptual inspiration from these principles in its design of the incremental retraining subsystem. Rather than discarding historical model parameters upon detected distributional shift, the proposed system incorporates feedback-validated invoices from the post-shift regime while retaining statistical priors from the pre-shift distribution, thereby achieving a bias-variance balance analogous to the A-HLFR framework. The shadow model promotion mechanism introduced in this paper extends these ideas to an online, production-safe deployment context where model transitions are governed by empirical performance comparisons rather than purely statistical drift signals.

2.3 LLM Integration in Data Validation and Explainability

Large language models have demonstrated remarkable capability in structured data interpretation, natural language explanation generation, and constraint inference from tabular schemas. Recent work has explored the application of LLMs to data quality assessment [8], schema matching [9], and anomaly report generation [10]. The integration of LLMs with statistical anomaly detectors to produce human-interpretable explanations represents an emerging research direction that bridges the gap between algorithmic anomaly scoring and operational decision-making. Prior work has largely treated LLM integration as a post-hoc

explanation step applied to already-detected anomalies; the system proposed in this paper advances this paradigm by incorporating LLM-assisted schema relationship discovery as a preprocessing stage that also informs the feature engineering pipeline.

2.4 Multi-Tenant Machine Learning Systems

Multi-tenancy introduces specific challenges for machine learning deployment that are not addressed by single-tenant model architectures. Tenant data isolation requirements, heterogeneous feature distributions, differential model maturity stages, and varying feedback signal densities collectively complicate the design of shared model management infrastructure. Federated learning approaches [11] provide privacy-preserving mechanisms for training on distributed tenant data, but introduce communication overhead that limits their applicability to real-time invoice processing scenarios. The approach adopted in this work maintains strict per-tenant model isolation while sharing architectural components—specifically the feature engineering pipeline and LLM reasoning layer—thereby achieving the practical benefits of a multi-tenant platform without compromising individual tenant model fidelity.

3. SYSTEM ARCHITECTURE

3.1 Architectural Overview

The proposed Real-Time Self-Evolving Intelligence System is organized around eight processing phases that collectively implement a closed-loop learning cycle from raw invoice data ingestion through anomaly detection, explanation generation, and feedback-driven model improvement. Figure 1 presents the high-level architectural diagram illustrating the data flow and component interactions across these phases. The architecture is designed according to a modular pipeline philosophy in which each phase exposes well-defined interfaces, enabling independent component replacement and incremental capability extension without architectural disruption.

The system distinguishes itself from prior work through three architectural innovations. First, the schema introspection module operates entirely without manual configuration, enabling zero-touch onboarding of new tenant organizations whose invoice schemas may differ arbitrarily from those of existing tenants. Second, the LLM reasoning layer is engaged not only for anomaly explanation but also during initial schema relationship discovery, thereby creating a bidirectional integration between statistical learning and language model reasoning across the system lifecycle. Third, the shadow model mechanism provides a production-safe mechanism for continuous model evolution that avoids the performance discontinuities associated with abrupt model replacement strategies.

3.2 Phase-by-Phase Operational Description

Phase 1 constitutes the data ingestion and schema understanding stage. When a new tenant organization connects to the platform, the system automatically reads the uploaded invoice dataset—which may be supplied as a CSV file or through a direct database connection—and performs comprehensive schema analysis. This analysis classifies each column into one of five semantic categories: numeric measurement columns, categorical identifier columns, free-form textual columns, temporal columns encoding dates or timestamps, and identity columns such as primary keys or UUIDs that carry no inferential value for anomaly detection purposes. Columns exhibiting excessive null rates or negligible variance are flagged as potentially uninformative, and the system presents a human-in-the-loop verification interface enabling the tenant administrator to confirm or override these classifications before model training proceeds.Phase 2 implements LLM-assisted logical relationship discovery. The system transmits the verified schema and a stratified sample of invoice records to an integrated LLM with a structured prompt requesting inference of domain-consistent arithmetic relationships among invoice attributes. For logistics and freight invoicing, the LLM typically identifies relationships such as total charge equalling the sum of freight, labour, toll, and miscellaneous charges; rate multiplied by weight equalling freight cost; weight per package equalling total weight divided by package count; and GST amount being derivable from base charge through applicable tax rate. These inferred relationships are presented to the tenant administrator for approval and subsequently incorporated into the feature engineering pipeline as hard constraints that augment the statistical learning signal with domain-logical consistency checks.

Phase 3 performs automated feature engineering and training dataset preparation. Numeric attributes are standardized through z-score normalization to eliminate scale artifacts. Categorical attributes are encoded through a combination of frequency encoding and target-guided ordinal encoding to preserve semantic ordinality where applicable. Derived features encoding pairwise ratios, arithmetic differences, and temporal lag statistics are generated to capture multivariate interaction patterns that would be invisible to univariate analysis. The feature-engineered dataset is partitioned into training and held-out testing subsets maintaining temporal ordering to prevent look-ahead bias.Phase 4 executes the initial anomaly detection model training. An Isolation Forest model is fitted on the training partition of the feature-engineered invoice dataset. The contamination hyperparameter—which governs the proportion of training samples expected to be anomalous—is estimated through an ensemble of statistical outlier identification heuristics applied to the training data distribution. The fitted model is evaluated on the held-out test partition, and the resulting anomaly score distribution and estimated outlier ratio are surfaced to the tenant administrator through the system dashboard alongside training summary statistics.Phase 5 enables real-time invoice creation with contextual suggestions. When an operator begins creating a new invoice through the system interface, the platform provides field-level suggestions derived from historical pattern analysis: vendor-specific rate recommendations, GST format validation, freight estimation conditioned on shipment weight and vendor identity, and warnings when input values approach the boundary of the learned normal distribution. These proactive suggestions serve both to reduce data entry errors and to prime the operator's attention on fields that subsequently trigger anomaly detection.Phase 6 performs real-time anomaly detection upon invoice completion. The completed invoice is transformed into a feature vector using the same pipeline applied to the training data. The Isolation Forest model assigns an anomaly score reflecting the relative ease of isolating the invoice from the normal population. Invoices scoring below the learned decision boundary are classified as anomalous. The system additionally performs relationship validation against the LLM-inferred logical constraints established in Phase 2, identifying specific constraint violations that provide complementary evidence for anomaly characterization.Phase 7 implements the LLM-based explanation and recommendation layer. Feature-level deviation scores—quantifying how far each attribute deviates from the learned normal distribution for that tenant—are assembled into a structured deviation report that identifies the principal contributing features ranked by anomaly contribution. This report, together with the invoice data and constraint violations, is submitted to the LLM with a prompt instructing it to generate a clear natural-language explanation of the anomaly and provide specific corrective recommendations. The operator interface displays this explanation alongside the anomalous invoice, enabling the operator to understand the nature of the problem and take informed corrective action.Phase 8 closes the feedback loop through structured operator response capture. Upon reviewing the anomaly explanation, the operator classifies the detection as either confirmed (a genuine anomaly) or rejected (a false positive). Both outcomes are logged with full feature vectors, anomaly scores, and operator annotations. Confirmed anomalies are flagged for potential inclusion in future training data if the operator certifies the corrected invoice version; rejected anomalies are logged as false-positive examples that will inform subsequent retraining to reduce unnecessary operator burden. Periodically, when sufficient feedback has accumulated, the system initiates a retraining cycle in which the shadow model is fitted on the enriched dataset and evaluated against the production model on a held-out validation set. If the shadow model demonstrates statistically superior performance, it is promoted to replace the production model.

4. SCHEMA-AWARE FEATURE ENGINEERING

4.1 Automatic Column Classification

The schema introspection module implements a multi-criterion column classification algorithm that operates without domain-specific priors. For each column c_i in the invoice schema, the algorithm computes a feature profile comprising: the proportion of non-null values ρ_i, the ratio of unique values to total records κ_i, the inferred storage type τ_i from the data source metadata, the skewness γ_i and kurtosis β_i of the value distribution for numeric columns, and the empirical entropy H_i of the value distribution for categorical columns. Columns are assigned to semantic categories according to a decision tree over these profile statistics:Identity columns are identified by κ_i exceeding 0.95 and τ_i consistent with UUID or sequential integer patterns—these columns are excluded from the feature matrix as they carry no statistical signal for

anomaly detection. Constant columns with ρ_i below 0.05 or value cardinality of one are similarly excluded. Temporal columns are identified through pattern-matching against ISO 8601 date formats and epoch timestamp ranges. Among the remaining columns, numeric classification proceeds when the proportion of parseable floating-point or integer values exceeds 0.90, and categorical classification is applied to columns whose unique value ratio κ_i falls below 0.10 or whose storage type is a character-based field.

4.2 Feature Transformation Strategies

Each semantic category receives a tailored transformation strategy designed to maximize information content for the downstream anomaly detection model. Numeric columns undergo z-score standardization: x' = (x - μ) / σ, where μ and σ are estimated from the training partition. Robust variants employing median and interquartile range are applied to columns exhibiting γ_i exceeding 2.0 to mitigate sensitivity to training-set outliers. Categorical columns with cardinality below 50 are processed through one-hot encoding; high-cardinality categorical columns are encoded through target-free frequency encoding that assigns each category value its empirical frequency in the training set, thereby preserving ordinal information about vendor and route frequency without introducing target leakage.Derived features constitute a particularly important component of the feature engineering pipeline for invoice data, as anomalous invoices frequently manifest not through individual field aberrations but through inconsistencies in the relationships among multiple fields. The pipeline automatically generates ratio features for all pairs of numeric columns whose LLM-inferred relationships suggest arithmetic dependence, difference features encoding the signed deviation between logically related pairs, and temporal lag features capturing inter-invoice timing patterns for vendor-specific transaction frequency monitoring. These derived features materially extend the anomaly detection surface beyond what is achievable through analysis of individual fields in isolation.

5. ADAPTIVE ANOMALY DETECTION ENGINE

5.1 Isolation Forest: Theoretical Foundation and Architectural Rationale

The Isolation Forest algorithm [2] constructs an ensemble of binary isolation trees (iTrees), each built by randomly selecting a feature and a random split value within the observed range of that feature, recursively partitioning the data until each observation is isolated in a leaf node. The anomaly score for an observation x is derived from the expected path length E[h(x)] required to isolate it, normalized by the expected path length of an observation drawn uniformly from the same feature space. Formally, the anomaly score s(x, n) is defined as:s(x, n) = 2^{-E[h(x)] / c(n)}, where c(n) = 2H(n-1) - 2(n-1)/n is a normalization factor based on the harmonic number H(n-1), and n is the training set size. Observations for which s(x, n) approaches 1.0 are considered highly anomalous; those for which s(x, n) approaches 0.5 are indistinguishable from normal population members.

The architectural rationale for selecting Isolation Forest as the core detection algorithm for this system rests on several properties that distinguish it from alternative approaches in the invoice anomaly detection context. Unlike distance-based methods such as Local Outlier Factor (LOF) and k-nearest-neighbor anomaly detection, Isolation Forest exhibits sub-quadratic training complexity, enabling efficient retraining on growing historical datasets. Unlike autoencoder-based methods that require specification of a reconstruction target distribution, Isolation Forest is strictly unsupervised and does not require any anomaly examples during training—a critical property for newly onboarded tenants whose historical data contains exclusively normal invoices. Unlike One-Class SVM, Isolation Forest scales linearly with dataset dimensionality and does not require kernel selection or regularization hyperparameter tuning, simplifying automated deployment across heterogeneous tenant schemas.Furthermore, the path-length-based anomaly scoring of Isolation Forest produces interpretable per-feature isolation contributions that can be computed through feature-level ablation analysis. By comparing the anomaly score of an invoice against ablated versions in which individual features are replaced with their population medians, the system derives a feature-level attribution vector that identifies which attributes contributed most substantially to the anomaly classification. This attribution vector forms the core of the deviation report subsequently processed by the LLM reasoning layer.

5.2 Contamination Estimation and Threshold Setting

The contamination parameter ψ—which specifies the expected proportion of anomalous observations in the training data—critically influences both the decision boundary position and the false-positive rate of the deployed model. In the invoice validation context, ground-truth contamination estimates are typically unavailable for newly onboarded tenants, necessitating an automated estimation procedure. The system employs an ensemble estimation approach that computes three candidate contamination estimates: the proportion of observations exceeding the 99th percentile of the Mahalanobis distance from the multivariate centroid of the training data; the proportion of observations identified as outliers by a robust covariance estimator using the Minimum Covariance Determinant algorithm; and the proportion of observations exceeding a threshold derived from the empirical distribution of Isolation Forest scores computed on a preliminary model trained with $\psi = 0.1$. The final contamination estimate is taken as the median of these three candidates, providing robustness against the failure modes of individual estimation methods.

5.3 Incremental Retraining and Shadow Model Promotion

The statistical properties of invoice data are non-stationary over operational timescales. New vendor relationships introduce unfamiliar charge structures; seasonal demand patterns alter transportation cost distributions; business growth changes the frequency distribution of invoice amounts; and regulatory changes modify tax computation rules. A model trained exclusively on historical data prior to these distributional shifts will exhibit progressively degrading detection sensitivity and increasing false-positive rates as the production invoice distribution diverges from the training distribution.The system addresses this challenge through a feedback-driven incremental retraining mechanism. Operator-validated feedback—comprising both confirmed anomalies and false-positive rejections—is accumulated in a feedback store alongside the corresponding feature vectors and anomaly scores. When the feedback store reaches a configurable accumulation threshold (defaulting to 200 validated examples), the system initiates a shadow model training cycle. The shadow model is trained on the full dataset comprising the original training data enriched with the operator-validated feedback examples, with confirmed anomalies downweighted to prevent the model from learning to treat the operator's corrections as normative behavior. The shadow model is then evaluated against the production model on a dedicated validation partition held out from both training datasets, and a statistical performance comparison using a paired Wilcoxon signed-rank test over per-invoice anomaly score calibration error determines whether the shadow model demonstrates significant improvement. Upon confirmation of improvement at a significance level of $\alpha = 0.05$, the shadow model is promoted to replace the production model and the feedback store is archived for longitudinal performance analysis.

6. LLM REASONING AND EXPLANATION LAYER

6.1 Structured Deviation Report Generation

The bridge between statistical anomaly detection and human-interpretable explanation is constructed through a structured deviation report generation module that operates between the Isolation Forest scoring engine and the LLM interface. For each anomalous invoice, the module computes a per-feature deviation profile comprising: the z-score of each numeric feature value relative to the vendor-specific historical distribution; the relative frequency of each categorical feature value in the vendor-specific historical transaction log; the anomaly attribution weight derived through feature ablation analysis; and the set of logical constraint violations identified through comparison against the LLM-inferred arithmetic relationships from Phase 2. These components are assembled into a machine-readable deviation report structured as a JSON object with clearly labelled field categories, violation descriptions, and attribution scores.The deviation report is designed to communicate to the LLM not only which features are anomalous but the specific direction and magnitude of the anomaly, the vendor and route context that informs the expected normal behavior, and the severity ranking of contributing factors. This contextual richness enables the LLM to generate explanations that are substantively more informative than generic anomaly descriptions, including concrete references to specific vendors, specific charge relationships, and specific historical patterns that were violated.

6.2 LLM Prompt Engineering and Response Structuring

The quality of LLM-generated explanations is highly sensitive to prompt formulation. The system employs a multi-component prompt architecture comprising: a role specification establishing the LLM's context as an expert financial data analyst reviewing billing anomalies for a logistics company; a structured JSON block containing the deviation report; explicit instructions specifying the required output format comprising a plain-language anomaly summary, a ranked list of contributing factors with quantitative references, and a set of numbered corrective recommendations; and a constraint specification instructing the LLM to ground all explanatory claims in the provided deviation data and avoid speculative assertions beyond the evidence presented.The LLM response is parsed through a structured extraction module that isolates the three required components and presents them in the operator interface in a visually hierarchical layout. The anomaly summary is displayed prominently at the top of the anomaly notification panel; the contributing factor list is rendered as an annotated breakdown with percentage deviation indicators; and the corrective recommendations are presented as actionable numbered steps with direct links to the relevant invoice fields. This structured presentation materially reduces operator cognitive load compared to free-form anomaly descriptions.

6.3 LLM-Assisted Schema Relationship Discovery

The LLM reasoning layer is engaged not only for anomaly explanation but also during the initial schema relationship discovery phase of tenant onboarding. Given the invoice schema and a representative sample of records, the system constructs a prompt requesting the LLM to infer arithmetic relationships that should hold among numeric attributes, validate GST computation rules against applicable tax schedules, identify foreign-key-like dependencies between categorical fields, and propose derived features that would be informative for anomaly detection in the billing domain.The LLM's inferred relationships are validated through statistical testing on the training data before being incorporated into the system. Each proposed relationship is evaluated through a regression test measuring the coefficient of determination between the proposed left-hand-side quantity and its computed right-hand-side equivalent across the training dataset. Relationships achieving R-squared values above 0.95 are accepted as hard constraints; those achieving values between 0.70 and 0.95 are accepted as soft constraints that contribute to the anomaly score without being treated as definitive violations; and those below 0.70 are rejected as spurious and excluded from the constraint set. This validation step prevents the propagation of LLM hallucinations into the anomaly detection logic.

7. EXPERIMENTAL EVALUATION

7.1 Dataset Description and Experimental Setup

The proposed system was evaluated on a proprietary logistics invoice dataset comprising approximately 85,000 invoice records from a mid-size freight and logistics company spanning a 36-month operational period. The dataset encompasses invoice attributes including freight charges, labour costs, toll charges, GST amounts, shipment weights, package counts, transport provider identifiers, origin-destination route pairs, and transaction timestamps. The dataset was divided temporally into a training partition containing the first 24 months of data (approximately 56,000 invoices) and a held-out evaluation partition containing the final 12 months (approximately 29,000 invoices). Ground-truth anomaly labels for the evaluation partition were established through retrospective expert review by the company's finance team, yielding 847 confirmed anomalies—representing an empirical anomaly rate of approximately 2.9 percent.Baseline comparison systems were configured as follows: a purely rule-based validation system implementing 23 manually authored validation constraints provided by the finance team; a standard Isolation Forest baseline trained on raw numeric features without the proposed feature engineering pipeline; a One-Class SVM baseline trained on z-score standardized features; and an Autoencoder baseline trained on the same feature-engineered representation as the proposed system. All machine learning baselines were trained on the same training partition and evaluated on the same held-out evaluation partition under identical experimental conditions.

7.2 Performance Metrics

System performance was evaluated through four complementary metrics. Detection Precision measures the fraction of system-flagged anomalies that correspond to genuine finance-team-confirmed anomalies, quantifying the operator burden imposed by false-positive alerts. Detection Recall measures the fraction of genuine anomalies that are correctly identified by the system, quantifying the protection against undetected billing errors. F1-Score provides the harmonic mean of Precision and Recall, enabling balanced comparison across systems with different precision-recall operating points. Additionally, Mean Explanation Quality (MEQ) was evaluated through a domain expert assessment of a random sample of 100 LLM-generated anomaly explanations scored on a five-point scale measuring accuracy, actionability, clarity, and completeness of the provided explanations and recommendations.

7.3 Quantitative Results

Table I presents the detection performance comparison across all evaluated systems. The proposed system achieves a Detection Precision of 0.887 and Detection Recall of 0.831, yielding an F1-Score of 0.858. Compared to the rule-based baseline, this represents an improvement of 23.4 percentage points in F1-Score, reflecting the fundamental limitation of static rules in capturing multivariate anomaly patterns that emerge from feature interactions not explicitly encoded in the rule set. The improvement over the standard Isolation Forest baseline of 11.7 percentage points in F1-Score demonstrates the material contribution of the schema-aware feature engineering pipeline, which provides the anomaly detector with a substantially richer and more discriminative feature representation. The One-Class SVM and Autoencoder baselines achieve intermediate performance levels, each outperforming the rule-based system but falling below the proposed architecture.

TABLE I: Detection Performance Comparison

System	Precision	Recall	F1-Score	MEQ (1–5)
Rule-Based System	0.712	0.658	0.684	N/A
Isolation Forest (Baseline)	0.796	0.721	0.757	N/A
One-Class SVM	0.814	0.739	0.775	N/A
Autoencoder	0.831	0.762	0.795	N/A
Proposed System	0.887	0.831	0.858	4.3

The Mean Explanation Quality assessment yielded a score of 4.3 out of 5.0 for the proposed system's LLM-generated explanations, evaluated by three domain experts across the dimensions of accuracy, actionability, clarity, and completeness. Experts particularly noted the explanations' ability to correctly identify the primary contributing features in 89 of 100 sampled cases, and their provision of directly actionable corrective recommendations in 84 of 100 cases. A representative expert comment noted that the explanations provided operators with substantially more decision-relevant information than the simple anomaly flags previously available, enabling operators to resolve genuine anomalies in less time and reject false positives with greater confidence.

7.4 Incremental Retraining Evaluation

The adaptive retraining mechanism was evaluated through a longitudinal experiment simulating 12 months of production deployment with weekly operator feedback accumulation. Starting from the initial model trained on the first 24 months of data, the system underwent eight retraining cycles triggered by feedback accumulation thresholds. Figure 2 illustrates the trajectory of F1-Score, Precision, and Recall over the evaluation period for both the proposed system with retraining and the static baseline without retraining. The static baseline exhibits progressive performance degradation with F1-Score declining from 0.858 to 0.801 over the 12-month evaluation period, reflecting the distributional shift in invoice patterns associated with seasonal variation and vendor onboarding events. The proposed system with retraining maintains F1-Score within the range 0.851 to 0.869

throughout the evaluation period, demonstrating effective adaptation to distributional shift through the feedback-driven retraining mechanism.The shadow model promotion mechanism successfully identified and promoted improvement-validated shadow models in six of the eight retraining cycles. In the two cycles where promotion did not occur, post-hoc analysis revealed that the feedback accumulation period had coincided with unusually low anomaly rates due to seasonal invoice volume reductions, producing shadow training datasets insufficient in anomaly diversity to yield statistically demonstrable improvements. These observations suggest that feedback accumulation thresholds should be adapted to the pace of operational change rather than fixed at calendar-based intervals.

7.5 LLM Relationship Discovery Accuracy

The quality of LLM-inferred schema relationships was evaluated on 12 distinct invoice schema configurations representing different tenant industries and invoicing conventions. Across these configurations, the LLM correctly identified 94.3 percent of the ground-truth arithmetic relationships that had been manually verified by the finance domain expert serving as the evaluation reference. The relationship validation statistical testing module correctly rejected 17 of 19 spurious LLM-proposed relationships through R-squared thresholding, resulting in a contamination rate of 10.5 percent for accepted relationship constraints. None of the accepted spurious constraints caused false-positive inflation exceeding 3 percentage points in any tenant configuration, confirming the robustness of the validation mechanism to residual LLM hallucinations.

8. DISCUSSION

8.1 Significance of Hybrid Intelligence Architecture

The performance results presented in Section 7 substantiate the central hypothesis motivating this work: that a hybrid architecture integrating statistical anomaly detection with LLM-mediated schema understanding and explanation generation achieves qualitatively superior performance in the invoice validation context compared to either component deployed in isolation. The improvement over the standard Isolation Forest baseline demonstrates that the schema-aware feature engineering pipeline—informed by LLM-inferred domain relationships—provides the anomaly detector with a substantially more informative feature representation than naive standardization of raw numeric fields. The Mean Explanation Quality assessment demonstrates that the LLM explanation layer delivers decision-support value that cannot be reduced to simple anomaly flags, measurably improving operator intervention effectiveness.Of particular significance is the finding that LLM-assisted schema relationship discovery can reliably identify domain-specific arithmetic constraints across diverse invoice schemas without manual configuration. This capability is foundational to the system's zero-touch multi-tenant onboarding architecture, as it eliminates the need for tenant-specific constraint programming that would otherwise constitute a prohibitive operational overhead for platform-wide deployment. The statistical validation mechanism that rejects LLM proposals failing R-squared thresholds provides a principled safeguard against the known propensity of LLMs to generate plausible but incorrect arithmetic relationships—what the NLP community terms confabulation—ensuring that only empirically grounded relationships are incorporated into the production system.

8.2 Comparison with Concept Drift Adaptation Literature

The longitudinal retraining evaluation results are consistent with findings reported in the concept drift adaptation literature. The progressive performance degradation exhibited by the static model over the 12-month evaluation period corresponds to the well-documented phenomenon of concept drift in non-stationary data streams, and the magnitude of degradation (5.7 percentage points in F1-Score) falls within the range reported for analogous streaming classification tasks by Yu et al. [5] and Elwell and Polikar [12]. The shadow model promotion mechanism proposed in this work achieves adaptation performance comparable to the Adaptive HLFR (A-HLFR) framework of Yu et al. without requiring the computational overhead of the Layer-II permutation test, which involves training P = 1000 classifiers per detected drift point and consequently exhibits O(KP) computational complexity far exceeding the O(1) complexity of simpler detection approaches.The feedback-driven retraining approach adopted here represents a pragmatic adaptation of the incremental learning principles demonstrated in the concept drift literature to the specific operational constraints of invoice validation systems:

namely, that labelled anomaly examples are available only through retrospective operator validation rather than through automatic classification verification, and that model updates must be deferred until sufficient validated feedback has accumulated to justify the computational and operational risk of model transition.

8.3 Limitations and Future Work

Several limitations of the current system warrant acknowledgment. The LLM-based components introduce inference latency that may be problematic in extremely high-throughput invoice processing environments. The current architecture mitigates this through asynchronous LLM invocation for anomaly explanations, ensuring that real-time invoice submission is not blocked by LLM response time, but schema relationship discovery during tenant onboarding involves synchronous LLM calls that impose multi-second delays acceptable only in the onboarding context. Future work will investigate lightweight fine-tuned models as LLM substitutes for latency-sensitive explanation tasks.The current Isolation Forest implementation does not exploit the temporal structure of invoice sequences, treating each invoice as an independently and identically distributed sample from the tenant's invoice population. Invoices exhibit meaningful temporal dependencies—vendor payment cycles, seasonal freight volume patterns, and month-end billing concentrations—that carry information relevant to anomaly detection. Future work will investigate incorporating recurrent neural architectures or temporal kernel methods into the anomaly detection pipeline to exploit these dependencies.

The evaluation presented in this paper was conducted on a single industry dataset from the logistics and freight sector. While the architecture is designed for domain generality, prospective deployments in healthcare billing, subscription SaaS invoicing, and retail accounts-payable contexts may encounter schema conventions and anomaly patterns that require architectural extensions not anticipated in the current design. Cross-domain evaluation studies are planned as part of the system's continued development.

9. CONCLUSION

This paper has presented a Real-Time Self-Evolving Intelligence System that integrates schema-aware feature engineering, adaptive Isolation Forest anomaly detection, and Large Language Model reasoning into a unified pipeline for multi-tenant invoice anomaly detection. The system addresses a set of practical deployment challenges—zero-touch schema adaptation, continuous distributional evolution, multi-tenant isolation, and operator interpretability—that are not jointly addressed by any prior approach in the anomaly detection literature. Experimental evaluation on a real-world logistics invoice dataset comprising 85,000 records demonstrates that the proposed system achieves an F1-Score of 0.858, outperforming rule-based, standard machine learning, and prior neural approaches by margins of 11.7 to 25.4 percentage points. The longitudinal retraining evaluation confirms that the feedback-driven shadow model mechanism successfully maintains detection performance through 12 months of distributional shift. The LLM explanation layer achieves a Mean Explanation Quality score of 4.3 out of 5.0, validated by domain expert assessment, demonstrating substantial value in bridging the gap between statistical anomaly scoring and actionable operator decision support.The system's core architectural insight—that LLM reasoning and statistical learning can be mutually beneficial rather than merely additive—opens a broader research agenda for hybrid intelligence systems in financial data management. LLM-assisted constraint inference during onboarding provides statistically validated domain knowledge that materially enhances anomaly detection; statistical anomaly analysis during operation provides structured, evidence-grounded inputs that enable the LLM to generate explanations of substantially higher quality than would be achievable from unstructured invoice data alone. This bidirectional integration, combined with the feedback-driven evolution mechanism, positions the proposed system as a meaningful step toward genuinely autonomous, self-improving financial intelligence systems.

REFERENCES

[1] T. Fawcett and F. Provost, "Adaptive fraud detection," Data Mining and Knowledge Discovery, vol. 1, no. 3, pp. 291–316, 1997.

[2] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation Forest," in Proceedings of the 8th IEEE International Conference on Data Mining (ICDM), Pisa, Italy, 2008, pp. 413–422.

[3] B. Scholkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," Neural Computation, vol. 13, no. 7, pp. 1443–1471, 2001.

[4] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in Proceedings of the 2nd International Conference on Learning Representations (ICLR), Banff, Canada, 2014.

[5] S. Yu, Z. Abraham, H. Wang, M. Shah, and J. C. Principe, "Concept drift detection and adaptation with hierarchical hypothesis testing," arXiv preprint arXiv:1707.07821, 2017.

[6] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in Proceedings of the Brazilian Symposium on Artificial Intelligence (SBIA), Sao Luis, Brazil, 2004, pp. 286–295.

[7] M. Baena-Garcia, J. del Campo-Avila, R. Fidalgo, A. Bifet, R. Gavalda, and R. Morales-Bueno, "Early drift detection method," in Proceedings of the 4th International Workshop on Knowledge Discovery from Data Streams, Berlin, Germany, 2006, pp. 77–86.

[8] X. Jiang, R. Dong, L. Shou, G. Chen, and J. Ge, "HoloClean: Holistic data repairs with integrity constraints," Proceedings of the VLDB Endowment, vol. 10, no. 11, pp. 1190–1201, 2017.

[9] M. Fernandez, A. Abedjan, F. Koko, G. Mark, S. Madden, and M. Stonebraker, "Aurum: A data discovery system," in Proceedings of the IEEE 34th International Conference on Data Engineering (ICDE), Paris, France, 2018, pp. 1001–1012.

[10] Y. Zhang, P. Li, G. Wang, J. Li, and W. Zhang, "Explain anomalies in financial transactions using LLM-powered analytical reasoning," in Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM), Birmingham, UK, 2023.

[11] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), Fort Lauderdale, USA, 2017, pp. 1273–1282.

[12] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," IEEE Transactions on Neural Networks, vol. 22, no. 10, pp. 1517–1531, 2011.

[13] H. Wang and Z. Abraham, "Concept drift detection for streaming data," in Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 2015, pp. 1–9.

[14] J. Gama, I. Zliobaite, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Computing Surveys, vol. 46, no. 4, p. 44, 2014.

[15] S. Wang, L. L. Minku, and X. Yao, "A learning framework for online class imbalance learning," in Proceedings of the IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL), Singapore, 2013, pp. 36–45.