SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

AI-Driven Anomaly Detection In Peoplesoft Financial Systems

Pradeep Narayanan

University of California, Berkeley, USA

Abstract
Enterprise Resource Planning (ERP) products like Oracle PeopleSoft have customarily been the transactional systems of record within large enterprises with millions of financial transactions flowing through ERP general ledgers, accounts payable, procurement, and expense management modules. Controls and monitoring of financial transactions within ERP systems have customarily been rules-based, with transactional data run through rule sets to check against pre-defined limits. Although useful, these mechanisms are backward-looking and context-blind and are gradually failing to keep up in the face of high-level fraud, explosive data growth, and ever-increasing regulatory demands. This article focuses on artificial intelligence (AI) use in ERP financial management, concentrating on machine learning-based anomaly detection. Architecture, technology, organization, and market perspectives on this transformation are discussed, with Oracle PeopleSoft as a case study for enterprise application suites. Topics include unsupervised learning algorithms and temporal neural networks. Engineering the architecture of ERP data structures for real-time integration, ensuring explainability for audit requirements, and addressing market drivers for AI integrations within legacy financial environments. The conclusion of the article is that such AI-based anomaly detection goes beyond a technological shift and is a fundamental reorganization of the way financial trust, governance, and oversight are conducted in large organizations. Firms that resist this shift will increasingly face governance risk.

Keywords: Artificial Intelligence, Anomaly Detection, Enterprise Resource Planning, Oracle PeopleSoft, Financial Fraud Detection, Machine Learning.

1. Introduction: The Governance Crisis Hidden Inside Modern ERP

Every day the world over, thousands of organizations rely on ERP systems, massive networks of interlocked applications that simultaneously process payroll checks, vendor payments, journal entries, procurement orders, and employee reimbursements at an ever greater scale. For higher education, government, and health care, the most important sectors where Oracle PeopleSoft has made its deepest inroads, it's necessary. It is the infrastructure for institutional accountability, regulatory compliance, and fiduciary responsibility. Worldwide, over 6000 organizations use PeopleSoft to perform critical financial business functions where inappropriate management of data or transactions can be disastrous.

Yet there is a quiet and largely unacknowledged governance crisis at the heart of how most ERP environments manage financial risks. That is because the dominant model for transaction oversight—rule-based controls—was designed for a world of lower transaction volumes, simpler fraud patterns, and far less data complexity. These systems evaluate each transaction, asking a simple question: does this transaction violate a known rule? If so, flag it. Otherwise, just approve it and move on. This architecture made sense when the world of financial threats was finite and knowable. But no longer does.

As financial fraud has become more complex, insiders have used their inside knowledge of an organization's architecture to evade controls. Vendors and other employees have become adept at creating a situation where they are close to a trigger threshold. Period-end manipulation, ghost vendor schemes, and other coordinated multi-transaction patterns tend to evade rule-based systems' controls. The Association of Certified Fraud Examiners estimates that each year, companies lose 5% of annual revenues to occupational fraud, which is not practically preventable with rule-based ERP controls at an enterprise scale [5].

The reason is that the advent of AI, more specifically machine learning, offers a fundamentally different approach: rather than asking whether a transaction breaks a known rule, it would ask whether a transaction resembles the kind of transaction that this organization, this user, in this context, at this time, typically produces. The move from rule-matching to pattern-learning is more than a technical improvement. It is a model shift to financial management thinking and operationalization that advanced anomaly detection and adaptive learning systems are the future of financial management within the firm, moving away from hypothesis-driven controls toward empirically supported risk assessment [17].

We will detail the reorientation below, covering the architecture of AI-based anomaly detection, the machine learning algorithms applicable to financial ERP data, the integration with PeopleSoft, organizational and regulatory conditions, and the context of the market that can make possible financial AI adoption for enterprises. We will focus on the first three in this article.

Table 1: Global scale and financial impact of occupational fraud across organizations [5]

Fraud Scale & Cost Metric	Value
Total cases analyzed across 138 countries	1,921
Total losses across all reported cases	>$3.1 billion
Estimated share of annual revenue lost to fraud	5%
Projected global annual fraud loss (based on 2022 GWP)	>$5 trillion
Median loss per case	$145,000
Average (mean) loss per case	$1,662,000
Cases with losses ≥ $1 million	22%
Median fraud duration before detection	12 months
Organizations with anti-fraud controls already in place	82%

2. The Architecture of Financial Data in PeopleSoft: What AI Has to Work With

Before we can look at what is possible with a PeopleSoft environment, we need to understand the depth and complexity of the data a PeopleSoft environment creates to support financial transactions. The scope, dimensionality, and relational richness of ERP data are well matched to the position of an anomaly detection system. PeopleSoft, in other words, is well-endowed.

2.1 The Chartfield Architecture as Analytical Substrate

The chartfield system is the foundation of the accounting data model in PeopleSoft. It is a multi-dimensional code structure that gives organizational meaning to every financial transaction entered in the system. Every transaction must have a value for the five dimensions of Account (what), Department (where), Fund (how), Program (why), and Project (who), called chart fields. A more complex system may implement additional dimensions such as product, class, or budget reference, which provides additional chart fields to input data of a special type.

While this type of structure enables flexible aggregation and drill-down of conventional business reporting on the organizational hierarchy, for AI anomaly detection, it provides a much richer high-dimensional feature space for distinguishing between normal and abnormal behavior. An otherwise unpublished account code for a particular department with a fund type otherwise limited to use by sponsored research, posted at 2:47 a.m. on a Sunday morning, may be individually explicable. These data points, as characteristic profiles, can be detected by a model of a typical pattern of transactions as abnormal, even if there is no explicit rule that encodes that characteristic profile as suspicious.

This understanding is the key to using AI in ERP financial processes, namely, that the use of a high-dimensional contextual data space combined with algorithms that can learn distributional norms provides a detection capability that is categorically more powerful than threshold-based rules [6].

2.2 Transaction Tables and Data Volume

PeopleSoft organizes financial transactions into related sets of tables. The journal entry tables include the header table (JRNL_HEADER) to hold transaction-level metadata and the line table (JRNL_LN) to hold line-level accounting distributions by chartfield. Vendor payments are stored in the VOUCHER and VCHR_LINE tables, while reimbursements for employee expenses are stored in EX_SHEET_HDR and EX_SHEET_LINE. Purchase orders are composed of PO_HDR and PO_LINE.

Tens of millions of rows of data from many years of financial history exist at even a medium-sized university or government agency. This is precisely the sort of data machine learning works well with. Machine-learning algorithms, by contrast, can take advantage of ever-growing data volumes. A corpus of historical transactions teaches the machine learning algorithm what normal financial behavior might look like. Whereas a rule-based approach defines financial normality abstractly, a machine-learning algorithm defines it statistically, in terms of the transactions (and the calendar) of a particular institution and the users and vendors it serves [2].

2.3 Pathways for AI System Integration

PeopleSoft provides several mature architectural mechanisms to allow integration with external AI systems without customizations to core application code, which is a key consideration for enterprise deployments where customizations incur long-term maintenance risk and may cause vendor upgrade issues.

Component interfaces are PeopleSoft business objects exposed programmatically to external consumers to read transaction data or to use PeopleSoft validation logic. Integration Broker serves as an enterprise service bus, publishing PeopleSoft transaction events to external consumers asynchronously using service operations. This means that every time a user saves a journal entry or submits an expense report for approval, a message can be sent to an external scoring engine that can run trained models on the data and return a risk score, all within the approval workflow cycle, before posting the transaction.

The Data Distribution Framework, delivered in PeopleTools 8.58 and 8.59, implements the machine learning case more explicitly with a standard implementation for how to extract training data for an external machine learning engine, how to consume model predictions via REST APIs, and how to govern and deploy machine learning models within PeopleSoft [11]. This implies that Oracle regarded AI not as an optional add-on , but as the new infrastructure deliverable and a requirement of all modern-day ERP environments.

3. Why Customary Rule-Based Controls Are Structurally Insufficient

To understand the value of detection based on AI, remember that rule-based controls are not only limited but also structurally weak as a primary control of financial anomalies. This is not to overlook the uses of business rules, which have definite legitimate uses, such as for required fields, chart field combination policies, or approval limits of certain types. Thus, the idea is that rules alone, as the main anomaly detection mechanism, are insufficient in today's financial environments.

3.1 The Known-Pattern Problem

Rule-based controls are only designed to catch transactions that the designer has anticipated would be a problem. Each rule is an assertion made by the designer about how a problematic transaction might look, such as amounts over a certain threshold , or certain chartfield combinations. Yet the disadvantage of this hypothesis-based architecture is that it is operationally useless against outliers whose pattern has not been anticipated or studied closely by the attacker and, therefore, the detection system.

Financial fraud is also adaptive. Insider threats with deep knowledge of the control environment can reverse engineer the rules and design their attacks to remain beneath detection thresholds. Structuring , the crime of breaking down payments below a threshold requiring additional approval or reporting is a well-known topic in forensic accounting and financial intelligence literature [13]. The $10,000 threshold for additional approval that is exploited is explicit in the relevant law, explaining the series of $9,800 payments. The rule sees exactly what it was designed to see: individual transactions within policy, while with respect to a model of transaction sequences the whole pattern is fully visible.

3.2 The False Positive Crisis and Alert Fatigue

The most important counter-argument to rule-based systems is that they produce large numbers of false positives. Financial auditors have found that many of the alerts produced by a system are simply normal transactions that match the trigger conditions but are not genuinely abnormal in any meaningful sense of the

word. The average false positive rate of rule-based detection environments can be as high as 95%, which means the analyst needs to review nineteen legitimate transactions to find one suspicious one [5].

These are not merely one-off consequences; they are magnified and created by the passage of time in a way that is both quantifiable and irreversible. The reviewers faced an impossible task. The hundreds of false positives weekly that fraud analysts must sift through make them susceptible to alert fatigue, discussed in behavioral economics and organizational psychology as becoming desensitized to the alerting stimulus, which produces gradually less analysis. As a result, the cruel irony is that rules-based systems that are meant to help highlight anomalies are instead continuously teaching humans checking the results to ignore alerts (the same defect that lets fraud through).

Machine learning approaches fundamentally change the problem from making flag-or-approve decisions to generating a risk score based on how statistically different the activity is to the normal expected behavior learned from the system. As a result, response actions are better aligned with the risk profile of each transaction. For example, low-risk transactions pass a standard process, while medium-risk transactions require additional documentation. High-risk transactions are reviewed by specialists with full context of the transactions. Fewer alerts and a larger share of true positives reduce investigator fatigue [1].

Table 2: Relationship between fraud duration and median financial loss [5]

Fraud Duration	Median Loss	% of Cases
Less than 6 months	$30,000	31%
7–12 months	$111,000	21%
13–18 months	$200,000	9%
19–24 months	$208,000	14%
25–36 months	$250,000	11%
37–48 months	$650,000	4%
49–60 months	$849,000	5%
More than 60 months	$875,000	6%

3.3 The Context Blindness Problem

A $45,000 journal entry entered by a senior grants accountant at 11 PM the night before the end of the fiscal year is not equivalent to a $45,000 journal entry entered by a junior-most expense processor at 2 am on a Tuesday in March even if they comply with the same rules or neither of them does. Who, when, what kind of account, in what organizational setting, and at what point in the fiscal calendar are the essential ingredients of factually grounded anomaly detection?

Rule-based systems are context-blind. They do not differentiate between roles, transaction histories, temporal positions, or organizational contexts. In well-designed machine learning systems, however, context is the main currency. Each feature in a model's input vector is one dimension of context. The model encodes the statistical relationship between the context features and the financial normality of the institution [8]. Therefore, transactions that are normal in all context features except for a few at once are flagged because this constellation of context features was never found in practice, according to the model's learned parameters.

4. Machine Learning Methods for Financial Anomaly Detection

Machine learning is a wide set of algorithms. There are different classes or families of algorithms that have been tailored to detect different types of anomalies. A complete anomaly detection system for finance may necessarily take into account such a variety of algorithms.

4.1 The Labeled Data Problem and the Case for Unsupervised Learning

The first challenge to machine learning applications for financial fraud lies in the scarcity of labeled training data. Supervised learning algorithms learn from examples of both normal and anomalous behavior. In particular, a learned anomaly detector must be trained on examples of confirmed fraud. First, in most organizations, fraud events are historically low and often unreliable, and there may be too few data events to

train supervised models. For instance, there may be hundreds of thousands of legitimate transactions, but only a handful of transactions with confirmed fraud within the last few years.
Unsupervised learning algorithms can solve this problem by reframing it as a distribution learning task. Instead of learning from samples of non-fraud and fraud, they learn what a normal transaction looks like from the population and detect anomalous transactions as outliers of this distribution. This requires only a reasonable corpus of predominantly legitimate transactions, which every ERP environment possesses in abundance [7]. The unsupervised model is thus not merely a technical preference but also a practical necessity in the organizational contexts where ERP anomaly detection is most needed.

4.2 Isolation Forest: Anomaly Detection Through Structural Partitioning

It is easy to see why Isolation Forest is one of the most popular unsupervised anomaly detection methods in industry and even within the financial industry: substantial theoretical advantages; actual efficiency even when the dimensionality is high; a solid empirical track record on financial data documented in many works; and a nice conceptual model that is simple enough to be intelligible in audit environments where explainability is a strong requirement.
The motivation behind the algorithm is elegant. The normal points, being the most numerous, and lying close to each other due to their high statistical density, require many random partitions until a single point is left out. The anomalies, being less numerous and lying in small regions of the feature space, are separated out by fewer random partitions. Therefore, the anomaly score is the normalized measure of how easy it is to isolate the given transaction. Anomalous transactions require less partitioning to be isolated, whereas normal transactions require more partitioning to be separated [10].
The Isolation Forest can be applied to the PeopleSoft transaction data as a feature vector. The features include the transaction amount transformed by a logarithm, the number of journal lines in the journal, the posting hour and day of the week, the user's average by account type, fund code novelty, and a chartfield combinations frequency. The algorithm learns the geometry of normal transaction space in all these dimensions. A transaction that occurs in a sparse region of that multi-dimensional space, where few previous transactions lie, is likely to receive a high anomaly score, regardless of whether any individual feature violates a threshold. This approach is in contrast to rule-based systems that instead rely on feature-based thresholds, whereas anomaly detection is based on learned distributional geometry [3].

4.3 Autoencoders: Learning to Reconstruct Normality

Autoencoder neural networks provide another powerful alternative for anomaly detection, particularly in cases where complex nonlinear patterns not easily detectable with statistical approaches can be found in financial data.
An autoencoder is a neural network that is trained to map the input data to a lower-dimensional latent representation and then reconstruct the original data. When the dimensionality reduction is forced, the network must learn the regularities in the training data or the statistical structure from which the original input is constructed. However, such a network can only reproduce regular patterns well; unusual patterns not encountered during training would be essentially impossible for the network to generate because of its training method.
If an anomalous transaction is shown to a trained autoencoder, then it will produce an output that is a bad reconstruction of the input, since it was not trained on examples similar to this anomalous transaction. Thus, the anomaly score of a transaction can be determined by the amount of reconstruction error, typically calculated as the mean squared error. A sufficiently high reconstruction error means that the transaction is unlikely to be part of the learned normal distribution [16].
In case of general ledger anomaly detection, deep autoencoder-based architectures, comprising several hidden layers that can learn hierarchical and non-linear relationships among nodes, have shown promising performance in terms of recall and precision. In particular, deep autoencoders applied to large-scale accounting datasets have been found to have a recall exceeding 80% in synthetic anomaly injection scenarios at acceptable precision levels [16]. Furthermore, autoencoders can capture nonlinear relationships between the features and can therefore be used to find complex patterns that linear algorithms cannot.

4.4 LSTM Networks: When Sequence Is the Signal

In modern ERP systems, financial transactions are not one-off but should occur as part of a sequence of events governed by organizational workflows and business process logic. A purchase requisition should lead to a

purchase order, which should lead to goods receipt, which should lead to invoice matching, which should lead to payment. An expense report should lead to submission, which should lead to managerial approval, which should lead to accounting review, which should lead to payment. Deviations from these expected sequences can be signals that control is either being broken or overridden.

The architecture of LSTMs is designed to provide memory across arbitrary time intervals. This is not the case with most feedforward neural networks, which process inputs one at a time independently. In contrast, the memory state of an LSTM changes with each newly processed element, allowing them to learn dependencies over many timesteps [9]. When trained on normal transaction sequences from PeopleSoft workflow data, LSTM models can discover anomalous transaction sequences, such as skipped approvals, payments preceding goods receipts, journal entries that reverse other entries in anomalous timing patterns, and sudden changes in users' characteristic patterns of transaction sequences after months of stable behavior [12].

LSTM autoencoders combine the strengths of recurrent networks for processing sequences with the reconstruction error-based anomaly scoring provided by standard autoencoders. For ERPs, this is highly useful, since learning compressed representations of the typical workflows in a sequence of normal transactions is desirable. Anomalous sequences that differ from the model either structurally or temporally will instead produce a high reconstruction error , and will subsequently flag as an anomaly. Such an approach is specifically useful to detect collective anomalies, or patterns of manipulation that are distributed across different transactions over a long period of time, which cannot be detected using point-level or simple threshold-based anomalies [14].

4.5 The Ensemble Architecture: A Combination of Algorithms

No single algorithm is optimal for every type of financial anomaly. Isolation Forests are able to successfully detect high-dimensional point anomalies but fail to detect subtle temporal dependencies present in sequential data. Although autoencoders can capture detailed nonlinear relationships, their construction and training are paramount. LSTM networks excel in detecting sequential anomalies but are computationally demanding and require careful feature selection across the broad range of temporal scales across financial workflows.

A promising empirically supported best practice is ensemble modeling that combines the outputs of multiple algorithms into a single composite risk score. For example, the combination of the outputs from an autoencoder and an Isolation Forest achieves better detection performance on several benchmark financial datasets [3]. The combination of scores is a weighted average of the Isolation Forest anomaly score, autoencoder reconstruction error, and LSTM sequence anomaly score, which produces a stronger risk score than any of the individual methods. This gives higher risk scores to transactions identified as suspicious by multiple models, improving anomaly detection, though in some cases where only one model flags a transaction, a soft response (like requesting additional documentation) may be more appropriate than stopping the workflow.

Table 3: Core unsupervised machine learning algorithms applied to PeopleSoft ERP financial transaction data [3] [10] [12] [16]

Algorithm	Anomaly Type Targeted	Key Strength	Limitation
Isolation Forest	Point anomalies	High-dimensional efficiency	Weak on temporal sequences
Autoencoder	Complex nonlinear patterns	Reconstruction error scoring	Training sensitivity
LSTM Network	Sequential/workflow anomalies	Temporal memory across steps	Computationally intensive
Ensemble (combined)	Multi-type anomalies	Higher precision and recall	Integration complexity

5. Feature Engineering: Turning PeopleSoft Data into Learning Signals

Although ultimately the performance of a machine learning system depends on the features used to train it, in the case of financial applications of ERP systems, the transformation of raw transaction records into feature sets encoding anomalous behavior is a task requiring substantial domain knowledge and statistical understanding. It is, in the judgment of most practitioners and researchers, the most consequential stage of the entire implementation process [6].

5.1 Transactional Features

The most basic features derived from a transaction are typically the amounts, highly important but rarely existing in USD; instead, they vary by several orders of magnitude, from minor petty cash reimbursements to large inter-fund transfers. This imbalance creates a severely right-skewed feature distribution that violates many algorithms' assumptions. Log-transforming this range compresses the scale, making the feature tractable across algorithms.

Many other attributes of the structure of the transaction itself can be signals of construct (for example, the roundness of the amount, whether the amount is a suspiciously clean number such as exactly $5,000.00 or $10,000.00, since round numbers are very unlikely for real-world expenditure amounts [13]). The number of lines in a journal entry encodes structural complexity and is anomalous if far above or below the user's historical average for that type of entry. Whether a transaction involves a foreign currency, introducing further risk factors, is a separate modeling problem. Whether the entry originated from automated Accounts Payable matching or was included via manual journal entry or by importing from an external source system is a categorical feature that has differing risk implications.

5.2 Temporal Features

Time is one of the most information-rich contextual variables in ERP transaction data but has not been extensively exploited in rule-based systems. The posting hour of a transaction itself is a good discriminator, because a transaction posted at 2:47 am has a much different profile compared to a transaction with the same structure but posted at 10:15 am. The day of the week can signal different levels of concern: weekend transactions might be more alarming if the organization does not normally operate then.

Another potential indicator of financial statement fraud is the clustering of manual journals at the end of a fiscal month or year, which may be another indicator of earnings management or other period-end adjustments to results [6]. Encoding each transaction date in relation to the entire thirty-day period (0.033 on day one and 1.0 on day thirty) allows models to separate normal transaction clustering usually found at the end of a period from abnormal transaction clustering.

Usually, in transactional systems, time elapsed between posting date and transaction date is of interest because it may point to backdating and retroactive changes of information on the financial books. In continuous auditing systems, temporal features (in particular, features that compare the user behavior to their own past behavior) are one of the most discriminative features for behavioral anomaly detection [8].

5.3 User-level and behavioral features

Behavioral features are calculated for each financial system user as a norm to which transactions are compared. Each user behaves in characteristic ways over time, such as transaction amounts for roles, regular account codes, posting behavior, and approval chains for transaction types. Mean and standard deviation of amounts, number of accounts touched, frequency with which common fund codes are accessed, and average time to completion of a workflow are calculated for each user in a rolling window, thus enabling the anomaly detection system to score the transaction against the user's behavior rather than against a population average [8].

The $10,000 journal entry would be unremarkable if made by the senior financial analyst handling a large fund transfer but would be anomalous if made by an administrative assistant in the same department whose previous journal entries have been, on average, $340. Without end-user behavioral baselines, all transactions are treated equally. With baselines, the population outlier (the administrator's large entry) is flagged as such, and the analyst's routine (in that context) is handled normally, regardless of its relationship to the population transaction data.

5.4 Relational and Chartfield Features

Relational features capture the organizational context that is specific to the PeopleSoft chartfield structure, as in whether a transaction is being posted to a fund code never before accessed by this operator. Does that transaction invoke a department/account combination that has no history of being bundled in the same transaction? Was more than one step eliminated from the approval chain? These are relational features that cannot be encoded into transactional features.

In higher education institutions, chartfield-derived features can indicate the risk related to the transaction, and therefore sponsored research fund indicators, restricted gift fund indicators, student account flags, and federal compliance indicators are important features to retain in the feature set [15]. For example, a user without a sponsored project history charging an expense to a sponsored research fund during non-business hours would

have a combination of relational features indicating a greater probability of being an anomaly, despite its relative dollar amount.

6. Implementation Architecture: Building the AI-ERP Integration

Operationalizing machine learning capabilities within a production PeopleSoft environment for effective anomaly detection can be a complex task that requires balancing data architecture for advanced analytics, operational architecture for security and reliability, and preserving existing financial business processes.

6.1 Data Extraction and Training Pipeline

The extraction layer is the foundation of the integration architecture. Journal queries against the PeopleSoft financial tables JRNL_HEADER and JRNL_LN, VOUCHER and VCHR_LINE for vendor payments, EX_SHEET_HDR and EX_SHEET_LINE for expenses, and reference table queries for chartfield descriptions and user role attributes create the unprocessed data set for feature engineering and model training. The extraction must be performance sensitive: very large queries against production database tables can be costly; therefore, the extraction should be scheduled during off-peak hours or run against read replicas where these are available within the infrastructure.

Data preparation includes other preprocessing tasks as well. Sensitive transaction (e.g., vendor identifiers, individual names, social security or tax identification numbers) data must be pseudonymized before transferring the data to external ML environments by replacing sensitive identifiers with consistent surrogate identifiers (preserving the underlying relationship structure of the data). Because many fields in real-world ERP data can be optional and historic data may not be in a current coding system, there may be missing values that must be imputed without introducing systematic bias to the feature space [2].

The model is trained across the window of transactions in the training dataset, typically a two- to three-year window to capture seasonality, fiscal years, and sufficient time to track long-term user activity patterns for baseline definition. An ensemble model is trained based on Isolation Forest models to handle point anomalies, autoencoder networks for multi-point anomalies, and LSTM networks to determine sequential workflows. For hyperparameter tuning, held-out historical data is used [3].

6.2 Real-time scoring and workflow integration

For true predictive prevention, we would need to score the transactions when they are being posted or during the approval workflow. This is accomplished using PeopleSoft's Integration Broker service operations. Upon saving or submission of transactions by a user, event messages are published to an external scoring API that invokes the trained models and returns composite risk scores to the PeopleSoft workflow engine to be processed within the allowed business process latency.

The system must have a risk-proportional response. For example, it may be that transactions with a score below the low-risk threshold are processed with no further action, whereas transactions that fall in the moderate-risk range require documentation and/or an additional layer of human review. Transactions that score above a high-risk threshold can then be routed to a specialist financial crime team, along with the reasons for the high-risk score. This staged response preserves one of the strengths of ERP systems—high levels of efficiency from automation—while ensuring that human intervention, which is truly a scarce resource, is applied where it can add maximum value.

If deployed with PeopleSoft's Approval Workflow Engine, the AI scoring system can be designed to use existing business process rules to determine how an AI risk score is used as an additional approver in the approval workflow. No architectural change is needed to take advantage of the AI scoring system. They can configure the AI score to trigger a predetermined workflow path according to their risk appetite.

7. Explainability, Audit Compliance, and Regulatory Considerations

If an organization adopts an AI system within a financial governance context, there are requirements that typically go beyond mere technical performance: in a financial context, systems are subject to internal audits, external regulatory oversight, and fiduciary accountability. Their reasoning behavior as well as their output needs to be understandable.

7.1 The Explainability Imperative

When a transaction is flagged as abnormal and results in a workflow intervention from a machine learning model, the decision made must be explainable to the following: The decision made must be explainable to the end-user (so they understand why their transaction was flagged), the financial officer (so they have sufficient understanding when approving or rejecting the transaction), and the internal or external auditor (to ensure that the rationale for detection is logical and consistent rather than based on pattern matching).

In black-box models with only a numerical risk score output, without attribution of risk score to specific predictors, there are serious compliance challenges at both the organizational and, increasingly, regulatory levels as this requirement is being codified into new AI governance regimes. The principle of explainability requires advocacy of appropriate techniques to render the model's reasoning transparent [4].

SHAP (SHapley Additive exPlanations) values are a rigorous feature attribution method from game theory. SHAP values decompose each model prediction into a sum of feature contributions. In our ensemble model, SHAP may attribute 40% of an anomaly score to late posting hour, 30% to the fund code atypical for the user, and 20% to the transaction amount exceeding the average behavior of the user in the training set. This attribution can also be rendered in plain language form for non-technical reviewers, transforming the opaque score into an actionable explanation that can support review and audit defensibility [4].

Counterfactual explanations are a different type of explanation. Whereas an explanation tells the user why a transaction was assigned to a certain label, a counterfactual explanation states what a transaction would have needed to be assigned a different label. The note "This entry would not have been flagged if it had been submitted during business hours on a weekday" provides the originator with a specific next plan of action and also gives the audit teams an opportunity to evaluate behavioral plausibility.

7.2 Audit Trail Requirements

Any AI used in the financial sector must have full logs of every prediction, along with the feature vector, model versioning, training date, composite and component risk scores, and a detailed record of every human decision made. A record of the name of the officer making the decision and the date on which the decision was made must also be recorded. This documentation is critical in proving that the detection system was stable and traceable; that any investigation performed as a result of detections is substantiated; and in helping measure the system's performance over time for drift or degradation.

The native PeopleSoft audit logging tool can be extended through Application Engine processes and Integration Broker publications to log AI events in a structured manner so they can be monitored by internal audit teams and external examiners. The audit log must be immutable and protected from modification by any user, including system administrators, and considered an evidentiary record.

7.3 Model Governance and Continuing Monitoring

Machine learning (ML) models are generally trained on prior history in financial companies. The predictive accuracy of these models deteriorates over time due to shifts in the data distribution caused by factors like changes in business lines, people departures and movement, reorganization, policy changes, and changes in transactional behavior from one financial year to the next. One potential problem with models trained on data for a specific year is that the model may classify as anomalous patterns of data that have since become normal, or it may fail to classify as anomalous patterns of data that have occurred since the model was trained [8].

Model governance typically requires regular model retraining. Depending on how often there are changing financial environments, this could be monthly or quarterly. The model can be monitored on disparate metrics such as precision, recall, false positive rate, and anomaly score distribution. Fluctuations in these metrics can indicate concept drift and be an impetus for model retraining even outside the regular cadence. All versions of all models should be stored for audit purposes so that if a model was used for a transaction eighteen months ago, it can be retrieved.

8. The Market Transformation: AI's Broader Impact on the ERP Landscape

The integration of AI into ERP financial systems is no longer simply a technical development taking place in the backrooms of organizations , but an emerging transformational trend in the marketplace for enterprise software. This trend is reshaping the competitive landscape among ERP vendors, the expectations around financial governance capabilities that ERP systems should be expected to deliver, and the operationalization of legacy ERP systems.

8.1 Competitive Pressure for the Old ERP Vendors

Oracle PeopleSoft has led for decades in categories like higher education, state and local government, and healthcare. However, Oracle PeopleSoft faces newer competition from other cloud-native enterprise resource planning (ERP) stacks such as Workday, Oracle Fusion Cloud, and SAP S/4HANA, which integrate (rather than bolt on) native AI and ML directly into their latest-generation foundational technologies.

This places increased pressure on PeopleSoft because organizations considering moving to the cloud will weigh the cost and disruption of the migration against the differences between PeopleSoft's AI capabilities and cloud-native solutions. Oracle's response has been to invest in PeopleTools such as the Data Distribution Framework, enhancements to Integration Broker, and native ML model management capabilities to give PeopleSoft users the ability to leverage AI without needing to move the core solution to the cloud [11]. There is an increasing consensus in the market that AI capability is no longer an effective competitive differentiator.

8.2 The Fraud Detection Market and AI Adoption Curves

In recent years the market for financial fraud detection technology has been growing quickly, driven by increasing fraud loss, regulatory pressure from governments, and the improved return on investment of AI-based fraud detection systems over rule-based and manual review systems. The success of the implementation of anomaly detection systems using machine learning has resulted in substantial cost reductions in terms of both the loss suffered from fraud and the cost of conducting financial oversight [5].

The ROI is accelerating the adoption curve of an organization type that has historically lagged in adopting new technology—government, higher education, and healthcare—and where PeopleSoft has its biggest installed base. With an established financial ROI, greater regulatory expectation that companies proactively monitor for fraud, and the availability of pre-engineered anomaly detection solutions that can easily be integrated with PeopleSoft without wide-ranging custom development, the diffusion of AI anomaly detection across the PeopleSoft installed base is expected to accelerate [15].

8.3 Organizational Readiness and Change Management

However, the remaining issue is no longer a technical one. There is no barrier to deploying AI anomaly detection in PeopleSoft environments. However, organizations will need to ensure that they have the necessary people, process, and governance in place to deliver AI financial monitoring.

Three organizational capability gaps are most likely to impair successful AI deployments for financial governance: data quality/governance maturity, data science/business financial operations collaboration, and leadership understanding of the AI systems to make informed decisions about how to configure them during their deployment and what risks the organization is willing to accept. Organizations where financial data is encoded differently across business units, where data science and finance teams operate as silos, or where executive leadership perceives AI projects as technology implementations rather than deploying new financial governance capabilities, lag behind peers who address these organizational versus purely technical dimensions.

However, applying AI-based anomaly detection is as much a change management exercise as it is a data science exercise. Business domain experts who understand PeopleSoft financial applications, chartfield structures, organizational processes, and the business meaning of transaction patterns must team with data scientists, business process experts, and data engineers through model feature engineering, model validation, and active model oversight. The data scientist may build a technically elegant but unconstrained model, or the financial officer may create an operational control system without analytical underpinnings. To build detection systems that are statistically powerful and operationally relevant, these two perspectives must be coupled.

9. Practical implementation considerations and best practices

When deploying AI anomaly detection capabilities for PeopleSoft environments, there are several practical considerations identified in research and professional practice that are seen as particularly important for successful implementation.

9.1 Begin with High-Risk Transaction Types

Successful implementations do not try to deploy anomaly detection across all transaction types in a business at once. They instead focus on the transaction types that carry the most risk related to anomalies, that can incur the highest costs from undetected fraud, and that have the best prospects for a clear financial return from AI-based anomaly detection.

Manual journal entries are the highest risk for the ERP financial processes, as they are not generated as sub-ledger transactions are completed, and they are the one transaction type that will require manual operator intervention to execute. Manual journal entries are the primary means of advanced financial engineering through ERP systems. Research finds that manual general ledger entries are the best target for anomaly detection using deep learning, as automated journal entries have systematic process controls that limit manipulation opportunities in ways that manual entries do not [16]. It is best to begin with manual journal entries, then single-vendor payments, and then employee expenses as the first AI targets to develop organizational confidence and capabilities before wide-ranging deployment.

9.2 Address Data Quality as a Prerequisite

However, no ML can be better than the data it is trained on. PeopleSoft environments that have been in production for several years and have undergone numerous organizational changes, system migrations, and codebase changes amass serious issues like inconsistent chartfield coding, optional fields sparsely populated, and legacy records predating current business logic and migrated records from predecessor systems with incompatible coding conventions [2].

Data quality assessment and remediation should be performed before model building. Completeness (what percentage of key fields are populated), consistency (are identical transaction types coded consistently across business units and periods), and accuracy (are reference data used to improve transaction records, e.g., vendor master, user role assignments, chartfield definitions, current and trustworthy?) should be assessed at the outset. For data that cannot be imputed, categorical indicators should be used rather than assumptions that create systematic biases in the data.

9.3 Align Thresholds with Organizational Risk Appetite

Elements of the system, such as the risk score thresholds that determine the movement of transactions between normal processing, improved review, and management scrutiny categories, are organizational or governance decisions. Choosing a conservative threshold that results in too many transactions being sent to the improved review process causes alert fatigue, which is the very issue the AI deployment aims to resolve. Setting the threshold too permissively, flagging only rare or extreme values, may result in missing some suspected anomalies.

Threshold calibration should be a joint effort involving the data science group as well as financial management, internal audit, and compliance functions, taking into account the regulatory environment, risk appetite, operational capacity to pursue additional investigations, and the historical cost of undetected financial anomalies. Thresholds should not be seen as technical constants but instead as governance parameters that should be adapted over time as operational experience is gained and data on the risk/score relationship is amassed.

Conclusion: Toward a New Architecture of Financial Trust

Clever automation of ERP financial oversight is a commercial reality with a clear direction and foundation that is becoming an organizational imperative as fraud losses and prescriptive regulations continue to increase and the sheer volume of transactions makes manual, rules-based financial oversight no longer possible or sustainable at a reasonable cost.

However, Oracle PeopleSoft is a well-architected application platform. The integration broker, component interfaces, the data distribution framework, and the depth of data available through a multi-dimensional chartfield structure, wide-ranging transaction history and complex workflow sequences mean that the practical gap between being able to exploit the platform as an AI engine for financial governance and the currently available cloud-native competitors is not as wide as might be suggested in marketing literature. .

What the research literature, practitioner experience, and market evidence show is that organizations with machine learning-based anomaly detection in their ERP financial data not only have a better fraud detection capability; they have a fundamentally different relationship between their transaction data and their governance capability. With machine learning, they can learn from their own financial history in ways that rules cannot encode and provide contextualization and integrated analysis capabilities within multi-dimensional space that threshold comparisons cannot provide. They also help to protect human reviewer attention, a genuinely finite resource, from being used up on false positives, leaving real anomalies to get the attention they need and deserve.

Per the canonical survey on anomaly detection, the problem is not whether something is an anomaly , but whether the anomaly is important [7]. Rule-based attack detection systems can answer this question by matching against a rule set or rule list. Machine learning systems will answer it by consulting the detailed web of statistical relationships existing within the organization itself, and this is at the heart of AI in ERP financial governance.

The cost of investments in data architecture, model development, integration architecture, organizational capabilities, and active governance is real and non-trivial. However, the outcome, measured in terms of reduced fraud losses, reduced cost of investigating false positive alerts, a stronger audit position, and higher confidence in the quality of data used for financial reporting, is demonstrably positive for well-implemented cases. The organizations that recognize this earliest , and invest accordingly will set the standard of financial governance that others will subsequently be required to meet.

References

[1] Mohiuddin Ahmed, et al., "A survey of anomaly detection techniques in the financial domain," in Science Direct, Future Generation Computer Systems, 55, 278–288. 2016. Available: https://doi.org/10.1016/j.future.2015.01.001

[2] Khaled Gubran Al-Hashedi & Preethiga Magalingam, "Financial fraud detection applying data mining techniques: A comprehensive review from 2009 to 2019," in Science Direct. Computer Science Review, 40: 100402, 2021. Available: https://doi.org/10.1016/j.cosrev.2021.100402

[3] Mahmood K. Mohammed and Miklos Telek, "Anomaly detection using a combination of autoencoder and isolation forest," in Research Gate, Proceedings of the IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI) (pp. 215–220). IEEE. 2023 Available: https://www.researchgate.net/publication/369254015_Anomaly_Detection_using_combination_of_Autoencoder_and_Isolation_Forest

[4] Alejandro Barredo Arrieta, et al., "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities, and challenges toward responsible AI," in Science Direct. Information Fusion, 58, 82–115. 2020. Available: https://doi.org/10.1016/j.inffus.2019.12.012

[5] Association of Certified Fraud Examiners, "Occupational fraud 2024: A report to the nations," ACFE. 2024. Available: https://www.ivey.uwo.ca/media/kjljj5cy/2024-report-to-the-nations.pdf

[6] Alexander Bakumenko and Ahmed Elragal, "Detecting anomalies in financial data using machine learning algorithms," in MDPI, Systems, 10(5), 130. 2022. Available: https://doi.org/10.3390/systems10050130

[7] Varun Chandola, et al., "Anomaly detection: A survey," ACM Computing Surveys, 41(3), Article 15. 2009. Available: https://doi.org/10.1145/1541880.1541882

[8] Hamed Hemati, et al., "Continual learning for unsupervised anomaly detection in continuous auditing of financial accounting data," in arXiv, computer science, machine learning. 2021. https://doi.org/10.48550/arXiv.2112.13215

[9] Sepp Hochreiter & Jurgen Schmidhuber, "Long short-term memory," Neural Computation, 9(8), 1735–1780. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[10] Fei Tony Liu et al., "Isolation forest," in ResearchGate, 2009. IEEE. Available: https://www.researchgate.net/publication/224384174_Isolation_Forest

[11] Rahul Mahashabde, "Introduction to machine learning with PeopleTools 8.58," in Oracle Blogs. 2020. Available: https://blogs.oracle.com/peoplesoft/post/introduction-to-machine-learning-with-peopletools-858

[12] Pankaj Malhotra, et al., "LSTM-based encoder-decoder for multi-sensor anomaly detection," in Proceedings of the ICML 2016 Anomaly Detection Workshop. ICML. Available: https://arxiv.org/pdf/1607.00148

[13] E.W.T. Ngai, et al., "The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature," in Elsevier, Decision Support Systems, 50(3), 559–569. Available: https://scispace.com/pdf/the-application-of-data-mining-techniques-in-financial-fraud-2xy182g2eu.pdf

[14] Huu-Du Nguyen, et al., "Forecasting and anomaly detection approaches using LSTM and LSTM autoencoder techniques with the applications in supply chain management," in Research Gate, International Journal of Information Management, 57, 102282. 2020. Available: https://www.researchgate.net/publication/346401830

[15] Chandra Rawat, "Role of ERP modernization in digital transformation: PeopleSoft insight," International Journal of Computer Trends and Technology, 71(2), 61–67. 2023. Available: https://ijcttjournal.org/Volume-71%20Issue-2/IJCTT-V71I2P110.pdf

[16] Marco Schreyer, et al., "Detection of anomalies in large-scale accounting data using deep autoencoder networks," arXiv. 2018. Available: https://doi.org/10.48550/arXiv.1709.05254
[17] Jarrod West and Maumita Bhattacharya, "Intelligent financial fraud detection: A comprehensive review," Computers & Security, 57, 47–66. 2016 Available: https://chengzhaoxi.xyz/download/pdf/paper/fintech/Intelligent-financial-fraud-detection_A-comprehensive-review.pdf