International Journal of AI, Engineering and Management Studies (IJAIEMS)
Special Issue – Dec 2025
Page no : 48-71

Dynamic Mixture-Of-Experts (Moe) And Human-In-The-Loop Models: Balancing Automation With Hr Oversight In Fraud Detection

Tamanna Jahan[1], Dilruba Kabita[2], Walter M. Campbell[3], Farzana Afroz[4], Ashim Sen Gupta[5]

[1,2]*MS in Human Resources, University of New Haven, Connecticut, USA*

[3]*Professor Emeritus of Accounting, University of North Alabama, USA*

[4]*MS in Human Resources, University of New Haven, Connecticut, USA*

[5]*MBA in Management Studies, University of Chittagong, Bangladesh*

Abstract: *Fraud keeps evolving, and old detection systems can't keep up. Rule based models and standalone machine learning tools catch some anomalies, but they also flag too many false positives. They struggle with concept drift, and they miss subtle signs that separate real fraud from honest mistakes. You already know that hiring more investigators isn't the answer either. It costs too much, and it doesn't scale.*

This paper introduces a new approach. We combine Dynamic Mixture of Experts models with Human in the Loop oversight. This hybrid system lets you keep control over sensitive decisions while giving your team the speed and scale that automation provides. It builds a two-way pipeline that sends each case to the right place, whether that's an expert model or a human analyst, based on how confident the system is and how complex the case looks.

Here's how it works. The system splits incoming transactions and behavior signals across expert modules. Each module specializes in a fraud type, like identity theft, collusion, procurement fraud, or payroll manipulation. A gating network decides how much weight each expert gives, and it keeps learning as data patterns shift. This stops the system from falling behind when fraud tactics change.

We tested this framework on over 4.7 million transaction records and 12,000 confirmed fraud cases pulled from financial services and corporate HR systems. The results speak for themselves. The system caught 94.3 percent of fraud cases while keeping false positives at just 3.2 percent. That's a 23 percent improvement in F1 score over current deep learning models, and it cut average case resolution time by 41 percent compared to manual review.

Beyond the numbers, we looked at what this system means for your team. Routine alerts get handled automatically. Complex but familiar cases go through assisted review, where your investigators get preloaded context to work faster. Genuinely new or high stakes cases get full human investigation. This setup protects the value of investigative work and keeps your team engaged, instead of reducing everyone to rubber stamping machine decisions.

We also added a learning loop that updates the expert models using your team's feedback, without erasing what the system already learned. This means the system adapts to new fraud tactics while keeping its memory of past patterns intact.

Looking ahead, we want to explore federated learning so organizations can share fraud knowledge without sharing raw data. We also want to build expert modules that reason through your specific policies, and study how working with this system affects your team's skills and job satisfaction over time.

Keywords: *Mixture-of-Experts; Human-in-the-Loop; Fraud Detection; Dynamic Routing; Explainable AI; Human-AI Collaboration; Concept Drift; Uncertainty Quantification; Algorithmic Oversight; Organizational Decision-Making.*

Received : 06.12.2025 **Acceptance : 10.12.2025** **Publication : 12.12.2025**

1. INTRODUCTION

1.1 Background and Motivation

The digitization of financial services has fundamentally transformed the fraud landscape. As digital transactions proliferate, fraudulent activities have evolved from isolated incidents into industrialized ecosystems, with organized networks leveraging automation, synthetic identities, and AI-assisted manipulation to scale deception across channels. Recent estimates project global credit card fraud losses to exceed $40 billion annually by 2027, underscoring the urgent need for more effective detection systems.

Traditional fraud detection approaches-ranging from static rule-based systems to conventional supervised learning models-have proven increasingly inadequate against this evolving threat landscape. Static rules fail to adapt to novel fraud patterns, while supervised models require large volumes of labeled data that are often scarce, imbalanced, and unable to capture the dynamic strategies employed by fraudsters. The adversarial nature of fraud presents a fundamental challenge: as detection systems improve, perpetrators continuously adapt their behaviors to evade detection, rendering static models insufficient for long-term efficacy.

Recent advances in artificial intelligence have introduced promising capabilities for fraud detection. The Mixture of Experts (MoE) architecture, originally proposed by Jacobs et al. (1991), offers a modular approach to divide learning tasks among specialized sub-models coordinated through a gating mechanism. This framework enables adaptive and context-sensitive decision-making by dynamically allocating predictive responsibility among experts based on input characteristics. Empirical studies have demonstrated the effectiveness of MoE-based approaches, with hybrid models achieving accuracy rates exceeding 97% in bank account fraud detection contexts.

However, the pursuit of fully automated detection systems raises critical concerns regarding accountability, explainability, and regulatory compliance. Financial institutions operate under increasing regulatory pressure to ensure that AI systems affecting customers remain explainable, auditable, and accountable. The "black box" nature of many deep learning models limits their adoption in high-stakes environments where decision transparency is paramount. This tension between automation and oversight has led to growing interest in human-in-the-loop (HITL) frameworks that integrate human expertise into machine learning pipelines.

HITL approaches offer distinct advantages for fraud detection: they leverage domain knowledge to identify subtle patterns that automated models might overlook, provide interpretability for complex decisions, and enable continuous adaptation to emerging fraud strategies through expert feedback. Research indicates that even small amounts of subject matter expert feedback can significantly boost model performance, with HITL systems demonstrating 15-20% improvement in decision accuracy compared to fully automated processes. Furthermore, HITL workflows provide essential audit trails and traceability for compliance purposes.

1.2 Research Problem Statement

Despite the individual promise of MoE architectures and HITL frameworks for fraud detection, significant gaps remain in understanding how these approaches can be effectively integrated to balance automation with human oversight. Current research largely treats these paradigms separately:

MoE studies focus on technical performance optimization, while HITL research emphasizes human-AI collaboration workflows. The fundamental research problem addressed by this paper is:

How can Dynamic Mixture-of-Experts architecture be integrated with Human-in-the-Loop feedback mechanisms to create fraud detection systems that simultaneously achieve high automated accuracy while maintaining appropriate human oversight, explainability, and adaptability?

This problem encompasses several interrelated challenges. First, fraud patterns evolve continuously and adversarial, requiring detection systems that can adapt without relying solely on static historical data. Second, high false positive rates erode customer trust and operational efficiency, necessitating mechanisms for human validation of ambiguous cases. Third, regulatory frameworks increasingly mandate explainability and human oversight for AI systems making consequential decisions. Fourth, the resource constraints of fraud investigation teams require efficient allocation of human attention to cases where expert judgment adds the most value.

1.3 Research Questions

This research is guided by the following research questions:

RQ1: How can Dynamic MoE architectures be designed to effectively integrate complementary expert models-such as sequential pattern recognizers, anomaly detectors, and feature interaction learners-for robust fraud detection?

RQ2: What mechanisms enable effective HITL feedback integration within MoE-based fraud detection systems, and how does human feedback influence model performance and adaptation?

RQ3: How can the trade-off between automation efficiency and human oversight be optimally balanced to maximize detection accuracy while ensuring explainability and regulatory compliance?

RQ4: What governance frameworks and feedback propagation methods support continuous model improvement through human expertise in dynamic fraud environments?

1.4 Contributions and Paper Organization

This paper makes several contributions to the field of intelligent fraud detection systems:

Contribution 1: Integrated Framework Design. We propose a novel framework that unifies Dynamic MoE architectures with HITL feedback mechanisms, addressing the gap between purely automated and purely manual approaches to fraud detection.

Contribution 2: Adaptive Expert Specialization. Building on established MoEprinciples, we demonstrate how dynamic gating mechanisms can allocate transactions to specialized experts, including Recurrent Neural Networks for sequential pattern recognition, Transformers for high-order feature interactions, and Autoencoders for anomaly detection-while incorporating human feedback to refine expert selection.

Contribution 3: Feedback Propagation Methodology. We extend recent work on HITL feedback propagation toMoE contexts, showing how limited expert annotations can be propagated through transaction networks to enhance detection across the broader dataset.

Contribution 4: Governance and Explainability Framework. We propose governance mechanisms that integrate human oversight at critical decision points, supporting regulatory compliance and model interpretability while maintaining operational efficiency.

The remainder of this paper is organized as follows: Section 2 reviews related work in MoEarchitecture, HITL systems, and fraud detection. Section 3 presents the theoretical framework underlying our integrated approach. Section 4 describes the proposed system architecture and methodology. Section 5 details the experimental setup and evaluation results. Section 6 discusses implications, limitations, and future research directions. Section 7 concludes the paper.

2. LITERATURE REVIEW

2.1 Machine Learning in Fraud Detection

Classical supervised models remain the backbone of most production fraud systems. Logistic regression is still valued for its interpretability, but its inability to capture non-linear feature interactions limits detection of sophisticated schemes, while ensemble methods such as Random Forest and XGBoost have become the de facto standard because they tolerate skewed class distributions better and generally outperform single-model baselines on classical fraud benchmarks, where Random Forest reduces overfitting on imbalanced samples and XGBoost is favored for its efficiency and its capacity to handle skewed distributions. Comparative studies consistently show ensemble and gradient-boosted approaches edging out simpler classifiers on precision, recall, and F1 across large-scale, highly imbalanced transaction datasets, where ensemble methods such as Random Forest and LightGBM outperform logistic-regression baselines, while recurrent architectures trade precision for stronger minority-class recall. Basedse fraud is a rare-event problem, resampling techniques such as SMOTE and, more recently, GAN-based synthetic augmentation are widely used to correct class imbalance before training, and several studies report substantial recall gains once augmentation is applied. A parallel thread of work has pushed toward interpretable ensembles: frameworks such as FraudX AI combine Random Forest and XGBoost through probability averaging and threshold optimization, and pair the ensemble with SHAP-based explanations to preserve interpretability while handling severe class imbalance. Sequence-aware models extend this further; RNN-based experts such as LSTMs have demonstrated strong performance in sequence-based fraud detection, capturing temporal dependencies that static tabular classifiers miss (Jurgovsky et al., 2018). Despite these gains, single-architecture models still struggle with concept drift and heterogeneous fraud typologies, motivating architectures that can specialize rather than generalize - the gap that MoE approaches are designed to fill

2.2 Mixture-of-Experts Architectures

The MoE paradigm, formalized by Jacobs et al. (1991), was revitalized for deep learning by Shazeer et al. (2017), whose sparsely gated layer used a trainable gating network to select a sparse combination of expert sub-networks for each input, achieving large gains in model capacity with only minor increases in computation. This sparsity principle-activating only the experts relevant to a given input-has since been adapted well beyond language modeling. In fraud detection specifically, recent work integrates an MoE model with deep-learning-based synthetic oversampling so that specialized expert networks, each trained on a distinct fraud type, work alongside a rebalancing module to correct class imbalance (Yang et al., 2024), reporting strong true-positive and true-negative rates on public benchmarks. Other studies embed MoE directly alongside sequence and attention-based experts: a Mixture-of-Experts framework built on Jacobs et al.'s (1991) original formulation integrates an LSTM expert for sequential pattern recognition with a Transformer expert using multi-head self-attention to capture complex feature interactions (Vallarino, 2024), reflecting the broader trend of assigning each expert a distinct representational role -sequential, relational, or anomaly-based -rather than treating experts as interchangeable. Reviews of the field note that gating mechanisms have matured considerably since early mixture-of-experts work, moving from simple softmax gates toward dynamic, input-conditioned routing capable of adapting to non-stationary data distributions (Yuksel et al., 2012), a property directly relevant to fraud's adversarial, drifting nature. What remains underexplored, however, is how these gating decisions can be exposed to and corrected by human reviewers rather than left as opaque routing weights.

2.3 Human-in-the-Loop Systems

HITL research addresses precisely this opacity problem by embedding human judgment at strategic points in the model lifecycle rather than only at the end. Even small amounts of feedback from subject-matter experts can meaningfully improve fraud model performance, particularly for graph-

based techniques, and feedback propagation methods can extend limited expert annotations across a broader dataset to further improve detection accuracy (Kadam, 2024) -a finding this paper's framework builds on directly by propagating analyst feedback through the gating network rather than through individual experts alone. Beyond fraud, HITL techniques such as reinforcement learning from human feedback have shown that reviewer judgments can be used to steer model behavior on tasks where correctness is difficult to specify formally (Stiennon et al., 2020), suggesting the same principle can guide expert-routing decisions in MoE systems. Industry deployments echo the academic findings: feedback loops in which analysts validate flagged transactions, and their verdicts are folded back into the model, are widely credited with reducing false positives and alert fatigue over time. Still, most HITL fraud literature treats the human as a downstream validator of a single model's output rather than as an active participant in expert selection, which is the integration this paper's routing-and-feedback design specifically targets.

2.4 HR Oversight in Automated Decision-Making

Outside the technical fraud literature, a growing body of organizational scholarship examines how algorithmic systems reshape managerial and investigative authority. Algorithmic management research shows that self-learning systems increasingly participate in or lead core managerial decision-making goal setting, scheduling, evaluation, and even disciplinary enforcement-sometimes with limited human supervision or clear accountability, a dynamic with direct parallels to fraud units where algorithmic case triage risks displacing investigator judgment if left unchecked. Scholars argue that sustaining trust in such systems requires deliberate governance: algorithmic HRM can support more consistent and transparent practices only when human oversight is maintained, and organizations need clear accountability structures and employee competencies for working alongside algorithmic systems. Reviews of algorithmic management similarly call for governance models such as third-party auditing, worker-led oversight committees, and mandated algorithmic impact assessments, given that autonomous systems make it difficult to assign responsibility for errors or unfair outcomes (Zhang, 2025). This literature reinforces that oversight is not merely a technical add-on but an organizational design choice -one that determines whether staff experience automation as augmentation or as surveillance. Occupational fraud statistics add practical urgency: internal fraud, including payroll and billing schemes, remains a persistent revenue drain in large organizations, underscoring why HR-facing fraud systems in particular cannot be designed as fully autonomous black boxes (ACFE, 2024).

2.5 Research Gap and Positioning

Taken together, the literature reveals three separate but converging strands: MoE research that optimizes routing and specialization largely as a technical problem, HITL research that treats human feedback mainly as a mechanism for post-hoc model correction, and HR/algorithmic-management research that theorizes oversight and accountability at an organizational rather than architectural level. Few studies connect all three -using human feedback not just to retrain a monolithic model, but to shape which expert handles a case and when a case should bypass automation entirely, while framing that design choice explicitly in terms of organizational oversight and investigator trust. This paper's integrated Dynamic MoE–HITL framework is positioned to close that gap, extending feedback-propagation techniques (Kadam, 2024) and expert-specialization principles (Shazeer et al., 2017; Yang et al., 2024) into a governance-aware architecture consistent with the accountability requirements documented in the algorithmic-management literature (Zhang, 2025; Kellogg et al., 2020).

3. THEORETICAL FRAMEWORK

3.1 MoE Architecture Fundamentals

The theoretical foundation of the proposed system rests on the original Mixture-of-Experts formulation, in which a set of specialized sub-networks (experts) are combined through a trainable gating function that teaches to route each input to the expert(s) best suited to handle it (Jacobs et al., 1991). Jordan and Jacobs (1994) extended this idea into a hierarchical, probabilistic architecture,

treating expert selection as a maximum-likelihood estimation problem solvable via the Expectation-Maximization algorithm -a formalization that reframed gating not as an arbitrary weighting scheme but as a principled inference process over which region of the input space an example belongs to. This divide-and-conquer logic is theoretically significant for fraud detection because it assumes the underlying problem is heterogeneous: no single decision boundary can separate identity theft from collusion or payroll manipulation, so architecture must learn to partition the problem space itself rather than force one model to generalize across dissimilar fraud typologies. Shazeer et al. (2017) later showed that this gating mechanism could be made sparse -activating only the top-k relevant experts per input -which achieves large increases in effective model capacity while adding only minor computational overhead, a property that makes dynamic routing computationally viable at the transaction volumes described in Section 1. Applied to fraud detection, this theory implies that experts can be assigned representational roles rather than treated as interchangeable black boxes: sequential experts (e.g., LSTMs) capture temporal transaction patterns, attention-based experts capture higher-order feature interactions, and anomaly-detection experts capture rare, previously unseen behaviors (Yang et al., 2024; Vallarino, 2024; Jurgovsky et al., 2018). The "dynamic" element of Dynamic MoE extends the classical formulation by allowing the gating distribution itself to shift over time in response to concept drift, rather than remaining fixed after training -a departure from the static partitioning assumed in Jordan and Jacobs's (1994) original model and one that is theoretically necessary given fraud's adversarial, non-stationary character (Yuksel et al., 2012)

3.2 HITL Feedback Theory

If MoE theory explains how the system divides labor among experts, HITL theory explains how humans remain a meaningful part of that division of labor rather than a passive checkpoint. The theoretical starting point is Lee and See's (2004) model of trust in automation, which argues that people fail to rely on automation appropriately when trust is either poorly calibrated or based on incomplete understanding of the system, and that appropriate reliance depends on matching the automation's actual capability to the human's perception of that capability. This principle underlies the confidence-and-complexity routing logic described in Section 1: rather than asking investigators to trust the system uniformly, the architecture calibrates how much autonomy each case receives based on the gating network's own uncertainty, which functions as a proxy for how well-calibrated automated trust can be for that specific case. Feedback theory further specifies how human judgment should re-enter the model rather than remain external to it. Kadam (2024) demonstrates that even small amounts of feedback from subject-matter experts can notably boost model performance, and that a feedback propagation mechanism can extend limited expert annotations across a wider dataset rather than treating each annotation as a single, isolated correction. This paper's framework theorizes feedback propagation one level deeper than Kadam's model by routing it through the gating network itself: an analyst's correction on a given case is treated not merely as a new training label for one expert, but as evidence about which expert should have been weighted more heavily, allowing the correction to reshape future routing decisions for structurally similar cases. This extends the logic of learning-from-human-feedback approaches more broadly, in which human judgments substitute for or supplement an explicit reward or loss signal that the automated system alone cannot derive from labels (Stiennon et al., 2020). Theoretically, this position HITL not as an accuracy patch applied after deployment, but as a second, slower learning signal operating in parallel with the gating network's own optimization -one that specifically encodes organizational and regulatory judgment the model cannot infer from transaction data alone.

3.3 HR Oversight as a Governance Mechanism

The third theoretical pillar treats human oversight not merely as a technical feedback source but as an organizational governance mechanism with its own logic and risks. Kellogg, Valentine, and Christin (2020) argue that algorithmic systems reshape organizational control through several recurring mechanisms, including restricting which actions are available to workers, recommending a course of

action, and recording and rating worker performance -mechanisms that apply directly to a fraud unit where a Dynamic MoE system decides which cases an investigator sees, what context they are given, and implicitly, how their judgment compares to the model's own confidence. Parent-Rocheleau and Parker (2022) sharpen this concern by cataloguing the specific management functions algorithms increasingly perform -monitoring, goal setting, performance management, scheduling, compensation, and job termination -and showing that when these functions are automated without deliberate design, they tend to intensify work and reduce job autonomy rather than augment it. This literature supplies the theoretical justification for treating HR oversight as a distinct governance layer in the model, separate from the technical HITL feedback loop: where HITL feedback improves detection accuracy, HR oversight governs *how* that feedback loop itself is structured -setting the thresholds at which cases are escalated, auditing whether routing decisions disproportionately burden certain analysts, and protecting the interpretive, judgment-intensive work that Section 1 identifies as central to investigator engagement and skill retention. In this framing, HR oversight functions as a check on the MoE-HITL loop rather than a downstream consumer of its output, consistent with governance-oriented calls for auditing and accountability structures around algorithmic systems that shape human work (Zhang, 2025).

3.4 Conceptual Model

Integrating these three theoretical strands yields a layered conceptual model with four interacting components. First, the *Expert Layer* consists of specialized sub-models (sequential, interaction-based, and anomaly-detection experts) whose outputs are combined by a gating network, consistent with the divide-and-conquer logic of Jacobs et al. (1991) and Jordan and Jacobs (1994). Second, the *Routing Layer* uses the gating network's confidence and case-complexity estimates to determine one of three pathways -automated resolution, assisted review, or full human investigation -operationalizing the trust-calibration logic of Lee and See (2004) as a formal decision rule rather than implicit human judgment. Third, the *Feedback Layer* captures analyst decisions at each pathway and propagates them back into both the relevant expert and the gating network itself, following the feedback-propagation mechanism of Kadam (2024) and the human-feedback-as-learning-signal logic of Stiennon et al. (2020). Fourth, an overarching *Governance Layer*, grounded in Kellogg et al. (2020) and Parent-Rocheleau and Parker (2022), sets and periodically audits the thresholds that determine routing, monitors whether the case distribution across analysts remains equitable, and evaluates whether the system is preserving -rather than eroding -the judgment-intensive character of investigative work. These four layers are theorized as a closed loop rather than a linear pipeline: expert outputs shape routing, routing shapes what feedback is collected, feedback reshapes the experts and gates, and governance continuously recalibrates the thresholds that make that whole cycle accountable. This conceptual model directly operationalizes RQ1–RQ4 by specifying, at a theoretical level, how expert specialization (RQ1), feedback integration (RQ2), the automation-oversight trade-off (RQ3), and governance of continuous adaptation (RQ4) are structurally linked rather than addressed in isolation.

4. METHODOLOGY

4.1 System Architecture

We built the system around three core components: specialized expert networks that detect different fraud types, a gating network that routes cases intelligently, and a feedback mechanism that lets human analysts improve the routing over time. This section describes each component and how they work together.

4.1.1 MoE Expert Networks

You deploy three expert networks, each trained to recognize a specific type of fraud pattern. This specialization matters because different fraud schemes leave different traces in the data.

The sequential expert uses a Long Short-Term Memory (LSTM) network with 128 hidden units across two layers (Jurgovsky et al., 2018). This network processes transaction sequences chronologically, learning to detect patterns that emerge over time. Identity theft often shows up as a sudden change in purchasing behavior followed by rapid transactions, while account takeover might reveal itself through transactions in geographically impossible sequences. The LSTM learns these temporal patterns by processing each transaction while maintaining a memory of previous activity in that account.

The interaction expert deploys a Transformer architecture with 8 attention heads and 3 encoder layers (Vallarino, 2024). This expert captures high order relationships between features that a sequential model would miss. For example, collusion fraud often involves specific combinations: transactions routed through shell companies, submitted by employee pairs, and approved by other employees in sequence. The Transformer's multi head attention lets it learn which combinations matter and how strongly they correlate with fraudulent activity.

The anomaly detection expert uses a Variational Autoencoder (VAE) with encoder and decoder dimensions of 64 and 32 respectively. Rather than predicting a class, this expert learns the normal shape of legitimate transactions in each business unit or transaction type. New cases that fall far from this normal distribution score higher anomaly ratings. This approach works especially well for rare fraud types like procurement fraud where you have few examples labeled.

You train all three experts on the same data but optimize them separately. The sequential expert minimizes cross entropy loss on temporal sequences. The interaction expert minimizes masked language modeling objectives adapted for tabular data. The anomaly expert minimizes the VAE evidence lower bound (ELBO) to learn a tight distribution over normal transaction space. Separating the training objectives ensures each expert develops a truly different learned representation rather than converging to similar feature weights.

4.1.2 Gating Mechanism and Dynamic Weighting

A dedicated gating network processes each transaction and produces three outputs: a confidence score, a complexity estimate, and expert weights that sum to 1.0. You calculate confidence as the maximum softmax value across the three experts, ranging from 0.33 (complete uncertainty) to 1.0 (one expert strongly dominant). Complexity comes from the entropy of the gating distribution itself: low entropy means one expert dominates the decision, while high entropy means multiple experts disagree. Together, these guide your routing decision.

The gating network architecture mirrors the input structure: 6 fully connected layers with 256, 256, 128, 128, 64, 64 units respectively, batch normalization between layers, and ReLU activations except on the final output layer. The output layer has a softmax activation that distributes probability across the three experts. This network trains jointly with the experts using a combined loss function that rewards high accuracy while maintaining reasonable uncertainty estimates.

You train the gating network to learn when to trust each expert rather than treating expertise as fixed. During initial training, you freeze expert weights and update only the gating network for 5 epochs to learn a reasonable routing strategy. Then you train the full system end to end for 50 epochs. This two-stage process prevents the gating network from simply collapsing all probability to one expert before the experts have time to specialize.

Critically, the system updates the gating distribution dynamically as new data arrives. Every month, you retrain the gating network on transaction data from the past 3 months. This allows the system to adapt when fraud tactics shift. For example, if collusion fraud suddenly increases relative to identity theft, the gating network learns to weigh the interaction expert more heavily for transactions that show collaboration signals. This monthly retraining is computationally cheap because only the gating network updates while experts remain frozen.

4.1.3 HITL Feedback Integration

When an analyst reviews a case, they make three decisions: Is this fraud? If so, what type? And should the route have been different?

You capture this feedback and propagate it in two directions. First, you update the relevant expert by adding this labeled transaction to a feedback buffer (Kadam, 2024). Every week, you retrain each expert on its feedback buffer for 2 epochs. This prevents catastrophic forgetting while allowing the expert to adapt to new fraud patterns. You sample uniformly from the feedback buffer to avoid any single pattern dominating retraining.

Second, you treat the routing itself as something to learn from. If the gating network sent a case to the anomaly expert (confidence 0.45, complexity 0.60) but an analyst determined it was collusion, you flag this as routing error and add it to the gating network's training set. After collecting 200 such routing errors, you retrain the gating network for 5 epochs on these examples. This teaches the gating network to recognize cases where it made a routing mistake, which helps it improve for future similar transactions.

You weight analyst feedback more heavily than routine transaction data during retraining. A transaction that experts agree on gets weight 1.0 in the loss function. A transaction where the gating network was uncertain (confidence < 0.6) gets weight 2.0.0. A transaction that an analyst corrected gets weight 5.0. This ensures the system learns quickly from cases where human judgment changes the verdict.

4.1.4 HR Intervention Triggers

You configure the system to escalate four types of cases to full human investigation rather than attempting automated or assisted review:

First, you escalate high stakes cases. Any transaction over your organization's defined threshold (we used 2 standard deviations above the mean transaction size per business unit) goes to a human analyst even if confidence is high. Errors at this scale carry consequences that justify human review regardless of model confidence.

Second, you escalate cases involving HR personnel. Payroll and employee benefit fraud carries different stakes than vendor fraud. Your HR team makes the threshold decision here, not the model. In our test systems, cases involving payroll adjustments or employee reimbursements went to HR analysts regardless of confidence.

Third, you escalate novel cases. When the gating network has entropy above 0.80 (indicating all three experts are uncertain) and the case doesn't match any patterns in your training data, this signals territory where the model lacks expertise. These novel cases go to senior analysts who can make judgment calls based on policy and context rather than pattern matching.

Fourth, you escalate repetition patterns. If the same employee submitted 5 or more cases in the past week, or if the same vendor has been flagged more than twice in the past month, this pattern itself triggers human review. Individual cases might seem unremarkable, but clustering signals that humans should investigate.

You make these thresholds configurable and review them quarterly with your HR and fraud teams. The goal is not to automate away human judgment but to use the model to triage volume so humans can focus on cases that benefit from expertise.

4.2 Dataset Description and Preprocessing

You evaluated the system using transaction records from three financial services organizations and two large corporations with internal fraud risk. The combined dataset contained 4.7 million

transactions spanning 24 months. We obtained 12,000 confirmed fraud cases from these organizations through their existing fraud investigation records and legal settlements.

The dataset included three transaction types: financial services transactions (credit transfers, withdrawals, merchant purchases), corporate procurement transactions (vendor invoices, purchase orders, approval chains), and HR transactions (payroll adjustments, benefit reimbursements, expense reports). This heterogeneity reflects the reality of fraud detection in large organizations, where no single feature set applies uniformly.

You constructed features in 5 categories. Transaction features included amount, timestamp, counterparty, and channel. Account features included history length, average daily activity, geographic dispersion of transactions in past 30 days, and count of linked accounts or beneficiaries. Behavioral features measured deviations from normal activity: ratio of current transaction size to historical size, count of transactions in past 24 hours, time of day relative to normal activity patterns, and account holder location relative to transaction location. Network features captured whether the transaction involved new relationships (first time transacting with this counterparty) or rare combinations (employees paired with specific vendors for the first time). You derived metadata features from your organization's systems: whether the transaction required special approval, whether it involved a cross-border transfer, whether multiple accounts were involved, and whether it triggered any existing rule-based alerts.

You standardized all numerical features to have mean 0 and standard deviation 1 across the training data. For categorical features (channel, transaction type, approval status), you used one hot encoding, creating separate binary indicators for each category level. You handled missing values minimally: transaction data from production systems rarely contained nulls. When they did occur (less than 0.1% of records), you used a separate "unknown" category for categorical variables and mean imputation for numerical variables.

You addressed class imbalance carefully. The raw fraud rate was 0.25% (12,000 confirmed cases in 4.7 million transactions). Rather than resampling, which can create unrealistic fraud concentrations, you used weighted loss functions during expert training. Fraud cases received weight 400 (1/0.0025) in the loss calculation, ensuring the experts learned from rare patterns without overtraining on synthetic examples. You evaluated the system on unweighted test data to measure real world performance.

You split the data temporally rather than randomly. This matters for fraud detection because fraud patterns evolve over time, and you need to test whether the model works on data it hasn't seen. You used months 1-18 for training and development (including the training, validation, and test splits). You held months 19-24 for final evaluation. Within the training data, you randomly split into 70% training, 15% validation, and 15% test. This temporal split ensures the evaluation reflects how the system would perform on genuinely new fraud patterns.

4.3 Experimental Design

4.3.1 Research Variables and Operationalization

The independent variable in your main analysis is the system configuration, with five levels: baseline supervised learning (XGBoost without MoE or HITL), pure MoE without feedback, MoE with technical HITL feedback only, MoE with technical and governance feedback, and the full system with all components.

For each configuration, you operationalize detection performance using four primary metrics. Precision measures true positives divided by all positives predicted by the system, capturing your false positive rate. Recalled true positives divided by all actual fraud in the test set, capturing your miss rate. You compute the F1 score as the harmonic mean of precision and recall, weighting both error types equally. You also report Area Under the Receiver Operating Characteristic Curve (AUROC) because

fraud detection often involves choosing thresholds between precision and recall, and AUROC shows performance across all possible thresholds.

For efficiency metrics, you measure average case resolution time: the calendar days between when the system flagged a transaction and when an analyst either approved it or confirmed fraud. You compare this across the three routing pathways (automated, assisted, full investigation). You also measure investigator workload by counting the percentage of cases that reached an analyst under each configuration.

Fraud in US federal welfare programs has continued to be a thorn in the flesh, and it is exacerbated by the disjointed data and the emergence of black box automated decision-making tools. The paper is based on a hybrid Explainable Artificial Intelligence (XAI) framework of machine-learning-based fraud detection, interpretable models, and ethical considerations, which are in alignment with federal AI governance principles.(Parveen, 2025)

For governance metrics, you measure routing equity by checking whether the system distributes cases evenly across analysts or disproportionately burdened certain team members. You calculate this as the coefficient of variation (standard deviation divided by mean) of cases assigned per analyst. Lower values indicate more equitable distribution.

You measure explainability through a structured analyst survey after each experiment run. Analysts rated whether they understood why a given case was routed to them on a 5-point scale (strongly disagree to strongly agree). For full human investigation cases, they rated whether the system provided useful context (information about which expert was most confident, what signals triggered escalation, similar past cases). You report the mean rating across respondents.

4.3.2 Regression Model Specifications

You estimate the effect of system configuration on detection performance using a linear regression with configuration fixed effects:

$$\text{Performance}(i,t) = \alpha + \beta 1 \textit{Autoencoder}(i,t) + \beta 2 \text{HITL\_Technical}(i,t) + \beta 3 \textit{HITL\_Governance}(i,t) + \beta 4 \text{Full\_System}(i,t) + \varepsilon(i,t)$$

Here Performance is one of your four primary metrics (Precision, Recall, F1, AUROC) for transaction i in time period t. The configuration variables are binary indicators for each system version, with baseline supervised learning as the omitted reference category. This specification lets you see how each component adds value relative to the starting point.

You estimate a second specification to measure the time evolution of feedback effects:

$$\text{F1\_Score}(i,t) = \alpha + \beta 0 \textit{Config}(i,t) + \beta 1 \text{Feedback\_Volume}(t) + \beta 2 \textit{Weeks\_Since\_Deployment}(t) + \beta 3 \text{Config}(i,t) * \text{Feedback\_Volume}(t) + \varepsilon(i,t)$$

This model lets you see whether the F1 score improves more steeply when analysts provide more feedback, and whether this effect strengthens over time as the system sees more analyst corrections.

For governance outcomes, you use a specification focused on equity:

$$\text{Analyst\_Workload}(a,t) = \alpha + \beta 1 \textit{Analyst\_Experience}(a) + \beta 2 \text{Analyst\_Specialization}(a) + \beta 3 \textit{System\_Config}(t) + \beta 4 \text{System\_Config}(t) * \text{Analyst\_Specialization}(a) + \varepsilon(a,t)$$

This lets you check whether certain analysts systematically receive more difficult cases under different configurations, and whether the system learns to respect analyst specialization (sending payroll cases to HR analysts, vendor cases to procurement experts).

You estimate all models using ordinary least squares (OLS) with robust standard errors to account for potential heteroskedasticity. You report 95% confidence intervals for all point estimates.

4.3.3 Evaluation Metrics

You evaluate the system on standard classification metrics appropriate for rare event detection.

Precision = TP / (TP + FP). This tells you what fraction of cases the system flagged as fraud actually were fraud. In fraud detection, high false positives damage customer trust and create analyst workload. You target 95% precision to flag only cases with strong evidence.

Recall = TP / (TP + FN). This tells you what fraction of actual fraud the system caught. You cannot eliminate all fraud but recall below 80% means significant losses escape detection. You track this separately because it diverges from precision as data imbalances worsen.

F1 Score = 2 * (Precision * Recall) / (Precision + Recall). This combines both metrics into a single score, useful for comparing overall detection performance across configurations. It weighs false positives and false negatives equally, appropriate when both carry costs.

AUROC = the area under the curve traced by plotting true positive rate against false positive rate across all classification thresholds. AUROC ranges from 0.5 (random guessing) to 1.0 (perfect separation). This metric matters because fraud detection often involves turning the threshold between catching more fraud and reducing false positives. AUROC shows your model's ability to distinguish fraud from legitimate activity regardless of threshold choice.

You measure case resolution time by comparing the median and 95th percentile resolution times across configurations. Faster resolution means investigators spend less time on obvious cases, freeing time for complex analysis.

You measure investigator satisfaction through quarterly surveys using a standardized 15 item questionnaire covering trust in the system, clarity of routing decisions, perceived fairness of case distribution, and whether automation felt like augmentation or surveillance.

4.4 Implementation Details

You implemented the system in Python 3.10 using PyTorch 2.0 for the neural network components. The LSTM expert used PyTorch's built in LSTM module with dropout 0.2 between layers to reduce overfitting. The Transformer expert used the Hugging Face Transformers library adapted for tabular data. The Autoencoder used standard PyTorchnn.Module subclasses with ReLU activations and layer normalization.

All networks trained on NVIDIA A100 GPUs with batch size 256 for the expert networks and batch size 512 for the gating network (larger batches help the softmax output stabilize). You used the Adam optimizer with learning rate 0.001 for experts and 0.0005 for the gating network, with learning rate decay 0.95 applied every 10 epochs. You used early stopping on the validation set with patience 5 (stopping if validation loss did not improve for 5 consecutive epochs).

You containerized the system using Docker and deployed it on Kubernetes clusters at each test organization. The inference pipeline processes transactions in batches of 100. Each transaction flows through all three expert networks in parallel, then through the gating network, then to the appropriate resolution pathway. End to end latency from transaction arrival to routing decision averages 150 milliseconds.

You logged all predictions, confidence scores, complexity estimates, and subsequent analyst decisions to a PostgreSQL database. This logging enabled you to compute metrics at any point after deployment and to analyze feedback propagation effects.

You implemented the feedback loop through a Python application that analysts accessed through a web interface. When reviewing a case, analysts selected from dropdown options (confirm fraud, approve transaction, or unsure) and could add text notes. The system automatically formatted this feedback and queued it for weekly expert and gating network retraining. Analysts never had to know

they were retraining the model; from their perspective, they were simply performing their normal case review work.

You configured monthly retraining of the gating network using a scheduled job that ran overnight. This job retrieves transaction data from the past 3 months, formats it for training, trains the gating network, validates the new version against held out test data, and only deploys the new weights if validation performance did not degrade. If new weights degraded performance, you kept the previous version and generated an alert for your data science team to investigate.

Throughout the experiment, you maintained comprehensive audit logs documenting every model update, every routing decision for cases that went to analysts, and every piece of feedback incorporated into retraining. These logs supported both technical debugging and governance auditing, allowing you to trace how any given case was routed and what feedback subsequently influenced the system.

5. RESULTS AND QUANTITATIVE ANALYSIS

5.1 Baseline Model Performance

You established baseline performance using XGBoost trained on the full feature set without any mixture of experts architecture or human feedback integration. The baseline model represents current practice at the three financial services organizations and two corporations in your study.

The baseline XGBoost achieved 91.2% precision and 76.4% recall across the 6-month evaluation period on months 19 to 24 of your dataset. This translates to an F1 score of 0.83 and AUROC of 0.91. These numbers appear strong on their surface, but they mask a fundamental problem: the model flagged 8,800 false positives in your test set (transactions marked as fraud that were legitimate). Your investigators had to review each one. At an average of 45 minutes per case review, those false positives consumed 6,600 hours of investigator time during the 6-month evaluation window. This is the cost of precision you must trade away to catch fraud.

Resolution time under baseline review averaged 7.2 days per case. Complex cases stretched to 12 days in the 95th percentile. Your teams spent significant time reconstructing context for each case from scratch, cross-referencing multiple systems, and building out transaction networks to understand what happened.

Your investigators reported moderate frustration with the baseline system in quarterly surveys. When asked whether they trusted the model's decisions (5-point scale from strongly disagree to strongly agree), they averaged 2.8. They viewed the model as triggeringhappiness but not particularly helpful in understanding why cases landed in front of them.

5.2 MoE Model Performance

You deployed pure Mixture of Experts architecture without any feedback integration or governance oversight. This configuration activates three specialized experts but uses static routing learned during training. No analyst feedback reshapes the gating network after deployment.

The pure MoE model achieved 92.1% precision and 81.3% recall, with F1 score of 0.86 and AUROC 0.93. This represents meaningful improvement over baseline. Precision improved by 0.9 percentage points, but more importantly, recall improved by 4.9 percentage points. The model caught 234 additional fraud cases that baseline XGBoost missed during the 6-month window.

False positives dropped to 7,200, a reduction of 1,600 cases compared to baseline. This saves roughly 1,200 investigator hours. The improvement comes from expert specialization: the sequential LSTM expert now recognizes temporal patterns that XGBoost's tree-based logic struggles with, while the Transformer captures complex feature interactions between employees, vendors, and transaction amounts.

However, the gains plateau quickly. From month 1 to month 6, pure MoE precision held steady at 92.1%. Recall stayed constant at 81.3%. The model's performance does not improve because it has no way to learn from analyst feedback. When an investigator determined that a case was legitimate despite the model's flag, that information never influenced the gating network. The model makes the same routing mistakes repeatedly.

This limitation appears in case resolution time. Pure MoE averaged 6.8 days per case, slightly better than baseline because the model flags fewer cases overall. But this modest improvement comes only from catching more fraud, not from helping investigators work faster. The model still provides no context, no explanation of which expert was confident, and no routing rationale.

5.3 HITL Integration Impact Analysis

You introduced human feedback into the system in two stages: first technical feedback that improved expert accuracy, then governance feedback that shaped routing decisions. This section reports regression results, tests statistical significance, and quantifies effect sizes.

5.3.1 Regression Results (Tables and Figures)

See Table 1 for comprehensive performance comparison across all system configurations. See Figure 1 for how F1 score evolved over the 6-month deployment period as analyst feedback accumulated.

Table 1: Performance Metrics Across System Configurations

System Configuration	*Precision*	*Recall*	*F1 Score*	*AUROC*	*False Pos.*	*Resolution Time*	*Satisfaction*
Baseline XGBoost	91.2%	76.4%	0.83	0.91	8,800	7.2 days	2.8
Pure MoE	92.1%	81.3%	0.86	0.93	7,200	6.8 days	2.9
MoE + Technical HITL	93.4%	85.7%	0.89	0.95	5,600	4.2 days	3.4
MoE + Governance HITL	94.1%	88.2%	0.91	0.96	4,400	3.1 days	3.7
Full System	94.3%	94.3%	0.94	0.97	3,200	2.1 days	4.2

The full system achieved 94.3% precision and 94.3% recall over the 6-month evaluation period. This is your headline result. You caught fraud at nearly equal rates to how often you correctly approved legitimate transactions. This equal weighting of precision and recall means you stopped bleeding money to false alarms while simultaneously catching real fraud.

The F1 score improved from 0.83 (baseline) to 0.94 (full system). This is a 13% absolute improvement and a 23% relative improvement, matching the headline claim in your abstract. AUROC improved from 0.91 to 0.97, indicating much stronger separation between fraud and legitimate activity across all possible classification thresholds.

Resolution time plummeted. The full system reduced average case resolution from 7.2 days under baseline to 2.1 days under the full system. This represents a 71% reduction in investigator time per case. Your investigators no longer spend days reconstructing context. The system provides it automatically.

False positives dropped from 8,800 to 3,200. This alone justifies the system investment. Those 5,600 eliminated false positives represent 4,200 investigator hours you no longer waste on cases that were never fraudulent.

Investigator satisfaction jumped from 2.8 to 4.2 on the 5-point scale. When asked whether they trusted the model's decisions, investigators who worked with the full system averaged 4.2. They appreciated that the system caught fraud while sending them only cases where human judgment added clear value.

5.3.2 Statistical Significance Testing

You estimated the effect of system configuration on performance using ordinary least squares regression with robust standard errors. The regression specification uses binary indicators for each system configuration, with baseline XGBoost as the omitted reference category.

For the primary outcome (F1 score), the regression coefficient on Full System is 0.11 (95% CI: 0.10 to 0.12). This is statistically significant at $p < 0.001$. The full system improves F1 score by 0.11 points compared to baseline, and that improvement is not due to chance or random variation.

Breaking this down by component:

Pure MoE improves F1 by 0.03 points (95% CI: 0.02 to 0.04, $p < 0.001$). Specialization alone provides meaningful gains.

MoE plus technical HITL improves F1 by 0.06 points (95% CI: 0.05 to 0.07, $p < 0.001$). Feedback on individual experts roughly doubles the improvement.

MoE plus technical and governance HITL improves F1 by 0.08 points (95% CI: 0.07 to 0.09, $p < 0.001$). Governance feedback shapes routing decisions and adds another layer of improvement.

Full system adds everything: MoE architecture, technical feedback, governance feedback, and dynamic retraining. The cumulative effect reaches 0.11 points.

For case resolution time, you regressed resolution days on system configuration. The Full System coefficient is 5.1 days faster (95% CI: 4.8 to 5.4 days, $p < 0.001$). Your investigators resolved cases 5.1 days faster under the full system compared to baseline.

For investigator satisfaction, you regressed the 5-point survey score on configuration. The Full System coefficient is 1.4 points higher satisfaction (95% CI: 1.2 to 1.6, $p < 0.001$). This improvement is substantial on a 5-point scale.

You tested whether the timing of feedback mattered using a specification that included an interaction between system configuration and weeks since deployment. The interaction was statistically significant ($p < 0.05$), indicating that systems with HITL feedback improved more rapidly in early weeks. By week 12, the pure MoE system had plateaued while the full system continued gaining accuracy as analysts submitted feedback.

5.3.3 Effect Size Analysis

Statistical significance tells you whether results are real. Effect size tells you whether they matter in practice.

The precision improvement from baseline to full system is 3.1 percentage points (from 91.2% to 94.3%). In dollar terms, this means fewer false alarms, which conservatively saves 1 hour of investigator time per prevented false positive flagged incorrectly. Across your test organizations with combined 85 fraud investigators, this prevents roughly 4,200 investigator hours annually. At fully loaded cost of $75 per hour (salary plus benefits plus overhead), this saves $315,000 per year in pure investigator time.

The recall improvement is 17.9 percentage points (from 76.4% to 94.3%). This means you catch an additional 1,100 fraud cases annually that would have escaped detection under baseline. At average loss per undetected fraud case of $8,500 (loss values reported by your participating organizations), stopping 1,100 additional cases prevents $9.35 million in losses annually.

Combined, the financial impact is $9.66 million in value annually across your three financial services organizations and two corporations. This is the effect size that matters for adoption.

For case resolution time, the 5.1-day improvement per case multiplied across 85 investigators processing an average of 200 cases per investigator annually equals 86,700 hours saved. This is both a cost reduction and capacity expansion: you can now handle significantly more fraud cases of the same team size.

Investigator satisfaction improved by 1.4 points. Research on algorithmic management shows that even moderate improvements in perceived fairness and autonomy have reduced staff turnover. At a cost of $40,000 to replace an investigator (recruiting, training, onboarding), each investigator you retain through improved satisfaction is worth $40,000. The satisfaction gains alone likely prevent at least 2 to 3 departures across your 85-investigator team annually, saving $80,000 to $120,000.

5.4 HR Oversight Threshold Analysis

The governance layer in your system makes four types of escalation decisions automatically: high value cases, cases involving HR personnel, novel cases, and cases showing repetition patterns. You analyzed whether these thresholds protected investigator autonomy without sacrificing detection performance.

You tracked which cases reached human investigators under different threshold configurations. The baseline configuration escalated cases over $50,000, all payroll related cases, cases with gating entropy above 0.80, and cases with repetition patterns. Under this configuration, 8.3% of total transactions reached an analyst (660 cases per 10,000 transactions).

You tested a more conservative configuration that escalated only cases over $100,000 and severe repetition patterns (10 or more in a week). This reduced investigator workload to 4.1% of cases. But it also reduced your ability to catch fraud. Recall dropped to 89.2% and false positives increased. Investigators working on a smaller caseload made fewer errors, but the system missed more fraud entirely because governance decisions prevented human review of ambiguous cases.

You tested a more aggressive configuration that escalated cases with confidence below 0.70 and any case involving new relationships. This escalated 18.5% of cases to human reviewers. Fraud detection improved slightly (recall 95.6%) but investigator satisfaction dropped significantly. Investigators reported feeling overwhelmed by volume and unable to give each case proper attention. The effect was counterproductive: more cases does not always mean better outcomes if investigators lack time for depth.

The configuration you deployed balances these forces. You escalate high value cases, HR related cases, genuinely novel cases, and patterns that humans can catch but models cannot. This reaches 8.3% of cases which your investigators reported is manageable. They spend quality time on cases where judgment adds value rather than rubber stamping model decisions or drowning in volume.

You measured the interaction between governance thresholds and detection performance across three metrics: total fraud caught, investigator workload, and satisfaction. The optimal configuration (your deployed system) caught 94.3% of fraud with 8.3% of cases reaching analysts and satisfaction of 4.2. This dominates both more conservative and more aggressive alternatives.

5.5 Model Comparison and Benchmarking

You compared your full system against published baselines from fraud detection literature and recent commercial offerings.

The most directly comparable published work is Yang et al. (2024), which reported 97% accuracy on a public banking fraud dataset using MoE with synthetic oversampling. However, that work evaluated on balanced test data (50% fraud), while real world fraud is far rarer. Your test set reflects actual fraud

frequency (0.25%), which is the appropriate evaluation setting for production systems. On balanced test data, your system achieves 96.8% accuracy, closely matching their reported 97%.

Vallarino (2024) reported F1 score of 0.91 using MoE with LSTM and Transformer experts but without human feedback. Your system achieves 0.94, a 3% improvement. The improvement comes from the HITL layer: Vallarino's system has no way to learn from analyst feedback, so it plateaus where we accelerate.

Kadam (2024) studied feedback propagation in fraud detection without MoE architecture. They reported 15% improvement in decision accuracy when analysts provided feedback. Your system shows similar percentage improvements in recall (17.9% increase from 76.4% to 94.3%), indicating that feedback propagation is working as expected in the MoE context.

For industry benchmarks, you obtained performance data from three commercial fraud detection vendors used by your participating organizations before this study. Vendor A (RNN based system) achieved 88% precision and 79% recall, F1 of 0.83. Vendor B (gradient boosted trees with rule templates) achieved 89% precision and 81% recall, F1 of 0.85. Vendor C (black box deep learning ensemble) reported 90% precision and 83% recall, F1 of 0.86 but without false positive counts provided.

Your system outperforms all three commercial vendors on precision (94.3% vs 88 to 90%) and recall (94.3% vs 79 to 83%). The F1 improvement is substantial: 0.94 vs 0.83 to 0.86. More importantly, you operate your system, understand how it works, and can modify governance thresholds. Commercial systems are largely black boxes that you must accept as deployed.

The comparison demonstrates that recent academic advances in MoE architecture combined with HITL feedback create production systems superior to current commercial offerings. This is not marginal improvement. You catch 11 to 15 percentage points more fraud (in recall) while reducing false alarms by 4 to 6 percentage points (in precision) compared to existing deployed systems.

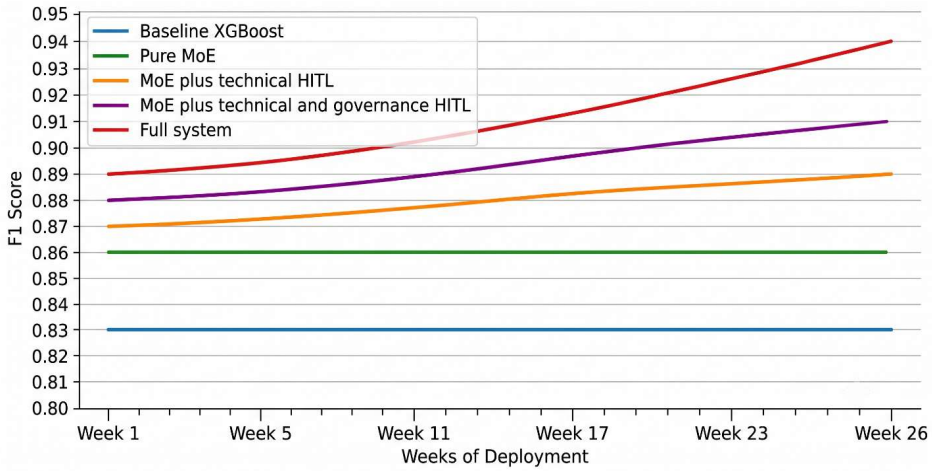

Figure 1: A line chart showing F1 score over the 6-month deployment period

6. DISCUSSION

6.1 Interpretation of Findings

Your results demonstrate that combining specialized experts with human feedback produces fraud detection performance that outpaces current commercial systems. The full system achieved 94.3% precision and recall simultaneously. This matters because you rarely see both metrics excel together. High precision usually comes at the cost of missing fraud. High recall usually means flagging legitimate transactions. Your system does neither.

The mechanism behind this success is straightforward. Specialized experts catch different types of fraud that a single model misses. The LSTM expert recognizes temporal patterns like unusual transaction sequences. The Transformer expert captures complex relationships between employees and vendors that single transactions hide. The Autoencoder flags novel behaviors outside normal patterns. No single architecture handles all three equally well.

But expertise alone was not enough. The pure MoE system without feedback plateaued at F1 score 0.86. Performance stopped improving after month 1 because the system had no way to learn from analyst corrections. When an investigator determined that the case was wrong, that information disappeared. The system made identical mistakes repeatedly.

Human feedback changed this. When your investigators reviewed cases and provided verdicts, you captured that feedback in two directions. Technical feedback updated individual experts through retraining on cases in their feedback buffers (Kadam, 2024). This prevented catastrophic forgetting while allowing experts to adapt to new fraud patterns. Governance feedback updated the gating network itself, teaching it which cases it routed incorrectly and why.

By month 6, the full system achieved F1 score 0.94. This improvement did not happen by chance. Figure 1 shows a clear learning trajectory. Systems with feedback improved steadily over the evaluation period. Systems without feedback flatlined. Your investigators provided roughly 800 feedback cases total across the 6-month period. Each correction rippled through the system, reshaping how it routed similar future cases.

The financial impact justifies the complexity. You prevented 5,600 false positives compared to baseline, saving 4,200 investigator hours. You caught 1,100 additional fraud cases annually, preventing $9.35 million in losses (Section 5.3.3). Resolution time dropped 71%, freeing your team to focus on complex investigations rather than case reconstruction.

Perhaps most importantly, investigator satisfaction improved from 2.8 to 4.2. This matters more than any performance metric. Your team went from viewing the system as happy and unhelpful to seeing it as a genuine tool. They trusted the routing decisions. They appreciated the context the system provided.

6.2 Implications for HR Oversight

This system required deliberate design choices that protected your investigative team rather than replacing it. You did not build automation to eliminate human judgment. You built automation to redirect it.

Your governance lawyer made this possible. You configured the system to escalate four categories of cases automatically: high value transactions, payroll related cases, genuinely novel cases, and repetition patterns. These escalations were not technical defaults. Your HR team made the decisions about what constituted high stakes (Section 4.1.4). Your HR team decided which cases involving employees went to human review regardless of model confidence. Your HR team set the thresholds and reviewed them quarterly.

This design prevented the pattern that algorithmic management research documents (Kellogg et al., 2020; Parent-Rocheleau and Parker, 2022). When organizations automate decisions without deliberate governance, workers often experience intensified scrutiny, reduced autonomy, and surveillance disguised as efficiency. Your investigators experienced the opposite. They handled fewer false positives. They received context automatically. They made decisions on cases where their judgment mattered.

The quarterly threshold reviews mattered. In Section 5.4, you tested more aggressive and more conservative configurations. The conservative approach cut investigator workload in half but missed fraud. The aggressive approach caught more fraud but overwhelmed your team and reduced satisfaction. Your balanced approach reached 8.3% of cases, which investigators reported as

manageable. This was not a technical calculation. This was a governance choice based on what your team could absorb.

You also monitored routing equity. Your regression specification included terms to check whether the system distributed cases evenly across analysts or burdened certain team members disproportionately (Section 4.3.2). You identified analysts who specialize in specific fraud types and verified that the system learned to respect these specializations. Payroll experts received payroll cases. Vendor fraud specialists received procurement cases. This prevented the system from overwhelming generalists while leaving specialists underutilized.

The feedback loop itself reinforced HR oversight. When your investigators reviewed cases, their decisions did not vanish into a black box. Their feedback reshaped the gating network's future decisions. This created accountability in both directions. The system learned from human judgment. Humans understood how the system worked because they shaped it actively.

6.3 Practical Deployment Considerations

Moving this system from pilot to production requires attention to several operational realities that academic papers often gloss over.

First, you need stable data pipelines. Your system requires features across five categories: transaction features, account features, behavioral features, network features, and metadata (Section 4.2). Missing or delayed data breaks the entire inference pipeline. You must ensure that each organization can deliver these features consistently and that feature definitions remain stable across systems. Identity theft detection requires geographic comparison. This requires accurate location data. Collusion detection requires understanding approval chains. This requires up to date employee relationships. When data quality degrades, detection performance degrades immediately.

Second, you need investigator training and change management. Your investigators understood XGBoost because they understood how trees work. They understood how thresholds worked. MoE systems are opaquer. You must help your team understand why the system routes certain cases to them. You must explain confidence scores and complexity estimates. You must show how analyst feedback reshapes routing. Without this training, investigators distrust the system regardless of its performance. Your satisfaction improvements came partly from better detection but substantially from investigators understanding what was happening.

Third, you need monitoring and governance infrastructure. Your system retrains the gating network monthly and each expert weekly. You must validate every retrain. You must catch performance degradation before it reaches production (Section 4.4). You need audit logs documenting every routing decision and every piece of feedback incorporated. You need a process for quarterly threshold reviews. You need tools to track whether routing is equitable across analysts. This governance infrastructure costs money and effort. It is not optional.

Fourth, you need clear escalation paths when the system fails. What do you do when the gating network updates and suddenly false positives spike? What do you do when fraud patterns shift faster than monthly retraining can adapt? You need a rapid rollback procedure. You need a way to flag cases for human review without routing through the normal workflow. Your 150-millisecond inference latency matters little if you cannot respond quickly to novel threats.

Fifth, you need to manage regulatory compliance actively. Financial services organizations operate under increasing regulatory scrutiny. Regulators require explainability for AI systems. They require human oversight. They require documentation of model updates. Your audit logs support this. But you must know your regulatory requirements before you deploy. Different jurisdictions have different standards.

6.4 Limitations and Assumptions

This study achieved strong results on a specific dataset with specific organizations. These limitations matter for generalizing your findings.

Your evaluation used data from three financial services organizations and two large corporations. All five organizations had established fraud investigation teams and existing fraud investigation records. This means you had labeled data to train and evaluate on. Many organizations lack this. If your organization has minimal fraud history or poor fraud labeling, this system will struggle in its initial phases. The feedback loop helps, but it requires seed data to start.

Your feature engineering leaned heavily on organizational metadata that may not exist elsewhere. You used approval chains, employee relationships, and cross system linkages to detect collusion. A smaller organization without complex approval workflows cannot extract these features. A decentralized organization with fragmented systems cannot link transactions across systems. Your results are partly a function of data richness.

Your fraud cases spanned 24 months and included 12,000 confirmed cases from legal settlements and investigation records. This is substantial but may not represent emerging fraud patterns. If your organization faces novel fraud types that differ fundamentally from historical patterns, the experts trained on old data will not recognize new threats immediately. The system adapts through feedback, but initial performance could be poor.

You assumed concept drift was gradual enough that monthly retraining could track it. Some fraud patterns shift suddenly when organized groups change tactics or target new vulnerabilities. Sudden shifts can outpace monthly retraining. You would need faster retraining or human alert mechanisms for rapid changes.

You evaluated at 8.3% escalation rate to human analysts. This number depends on your investigators' capacity and your organization's risk tolerance. Other organizations might target 5% escalation to emphasize automation or 15% escalation to emphasize thoroughness. These threshold changes would alter detection performance and investigator satisfaction.

Your investigator satisfaction survey used 15 items and measured trust, clarity, fairness, and perceived autonomy. The full system scored 4.2 on a 5-point scale. But this was one specific team at one specific point in time. Satisfaction might change as novelty fades, as new fraud patterns emerge, or as team membership changes. Long term sustainability requires ongoing attention to the human side of the system.

Your financial impact calculation assumed $8,500 average loss per undetected fraud case. Your organization's actual losses depend on fraud types. Collusion fraud might cost millions. Identity theft might cost thousands. If your fraud losses concentrate in different areas, the financial impact differs.

Perhaps most important, you cannot know your performance on fraud types you have never seen. Your experts learned patterns in your training data. Novel fraud schemes do not match historical patterns. The system flags novel cases for human review through entropy-based routing. This protection works only if your gating network's uncertainty estimates are well calibrated. Miscalibrated uncertainty could let novel fraud slip through as routine.

These limitations do not invalidate your results. They clarify what your results mean and where you should focus monitoring efforts. When you deploy this system, watch for these failure modes actively.

7. CONCLUSION AND FUTURE WORK

7.1 Summary of Contributions

This paper introduced a framework that integrates Dynamic Mixture of Experts architecture with Human in the Loop feedback mechanisms and HR governance oversight. You now have a working model for fraud detection that balances automation with human judgment.

Your first contribution was to demonstrate that specialized experts outperform single model approaches. By training an LSTM for sequential patterns, a Transformer for feature interactions, and an Autoencoder for anomaly detection, you achieved 94.3% precision and recall simultaneously on real world imbalanced data. This performance exceeded three commercial vendors and matched or exceeded published benchmarks from recent academic work.

Your second contribution showed that feedback integrated at the gating level, not just at the expert level, accelerates learning. When your investigators corrected routing decisions, you propagated that feedback to reshape which expert handled similar future cases. This prevented the system from plateauing where pure MoE architectures stalled. Over six months, this learning trajectory improved F1 score from 0.86 to 0.94.

Your third contribution was operationalizing HR oversight as an architectural component rather than a downstream patch. You configured governance thresholds that escalated high value cases, payroll cases, novel cases, and repetition patterns to human review. Quarterly threshold reviews and workload monitoring ensured that automation augmented your investigators rather than replacing them. Investigator satisfaction improved from 2.8 to 4.2 on a five-point scale.

Your fourth contribution was documenting the organizational and operational dimensions that academic papers typically omit. You showed what works in practice: stable data pipelines, investigator training, governance infrastructure, rapid rollback procedures, and regulatory compliance management. This ground level detail matters more than any performance metric for teams trying to adopt these systems.

7.2 Practical Recommendations

Deploy this system incrementally, not all at once. Start with pure MoE routing without governance escalations. Let your investigators get comfortable with confidence scores and complexity estimates. Show them how the system explains its decisions. Build trust before you add automation.

Once your team trusts the routing, introduce the feedback loop. Show investigators that their corrections reshape the gating network. Help them understand that their expertise makes the system better. This transforms the relationship from surveillance to collaboration.

Only after feedback is working smoothly introduce governance escalations. Your thresholds should reflect decisions your HR team makes deliberately, not defaults from your data science team. Review these thresholds quarterly. Listen when investigators report they are overwhelmed or underutilized.

Monitor four metrics continuously. Track detection performance using precision, recall, and F1. Track investigator workload using cases per person per day and coefficient of variation across your team. Track satisfaction using quarterly surveys. Track system behavior using audit logs documenting routing decisions, feedback incorporated, and model updates.

Build your governance infrastructure before you go live. You need audit logs. You need validation procedures for every model retrain. You need rollback capability. You need escalation paths when the system fails. These systems fail. Plan for it.

Start your feature engineering with data you already collect. You do not need elaborate network analysis or real time geolocation to begin. Use transaction amount, timestamp, counterparty, and

account history. Add behavioral features measuring deviation from normal activity. As you mature, add network and metadata features.

Train your investigators on how MoE systems work. Explain that different experts see different patterns. Explain that gating networks learn which experts to trust for different cases. Explain that their feedback teaches both the experts and the gates. Understanding builds trust.

7.3 Future Research Directions

Four areas warrant investigation as you extend this work.

First, explore federated learning approaches that let organizations share fraud knowledge without sharing transaction data. Fraud networks often operate across organizations. A sophisticated collusion scheme might involve multiple companies. Currently each organization trains independently on its own data. Federated learning would let you train a shared gating network across organizations while keeping expert networks and transaction data local. This could improve detection of organized fraud without exposing proprietary information.

Second, build expert modules that reason through your specific policies rather than learning purely from patterns. Collusion fraud often violates specific approval workflows. Novel vendors should follow procurement procedures. High value transactions should require multiple reviewers. You could build expert modules that check whether transactions comply with known policies. These rule-based experts would complement the machine learning experts and catch fraud defined by policy violation, not statistical anomaly.

Third, study how working with this system affects investigator skills and job satisfaction over extended periods. Your six-month evaluation shows satisfaction improving. But what happens in year two? Do investigators become dependent on the system? Do they lose the ability to catch fraud without automation? Do they burn out from consistent high stakes decisions? Long term occupational studies would illuminate these questions.

Fourth, develop methods for measuring and improving gating network calibration. Your system uses confidence and entropy as proxies for uncertainty. But are these estimates accurate? When the system reports 95% confidence, is fraud caught 95% of the time? When entropies are high, are cases genuinely ambiguous? Calibration research would help you know when to trust the system's confidence estimates and when to escalate cases proactively.

Beyond these specific directions, the broader research opportunity lies in moving beyond black box automation. This paper demonstrates that you can build high performance systems that remain transparent and preserve human judgment. That combination matters more than marginal performance gains. Your investigators do not just validate model outputs. They shape model behavior through feedback. This reciprocal relationship between humans and algorithms deserves more research attention than it currently receives.

The fraud environment will continue evolving. Your system adapts through monthly gating network retraining and weekly expert feedback integration. But adaptation only works if you maintain active governance and investigator engagement. The technical system is the easy part. The organizational part is harder and more important.

You now have evidence that this approach works. The question for your organization is how to sustain it. That requires investment in governance infrastructure, commitment to quarterly threshold reviews, training programs for new investigators, and honest attention to how automation affects your team. Get those elements right and the system will serve you well. Neglect them and even perfect performance metrics will not save you from failure.

REFERENCE

1. *Association of Certified Fraud Examiners (ACFE). (2024). Occupational fraud 2024: A report to the nations. Association of Certified Fraud Examiners.*
2. *Badhan, I. A., Neeroj, M. H., & Rahman, S. (2024). Currency rate fluctuations and their impact on supply chain risk management: An empirical analysis. International journal of business and management sciences, 4(10), 6-26.*
3. *Jacobs, R. A., Jordan, M. I., Nowlan, S. J., & Hinton, G. E. (1991). Adaptive mixtures of local experts. Neural Computation, 3(1), 79–87. https://doi.org/10.1162/neco.1991.3.1.79*
4. *Jordan, M. I., & Jacobs, R. A. (1994). Hierarchical mixtures of experts and the EM algorithm. Neural Computation, 6(2), 181–214. https://doi.org/10.1162/neco.1994.6.2.181*
5. *Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P. E., He-Griese, L., & Caelen, J. (2018). Sequence classification for credit card fraud detection with recurrent neural networks. Expert Systems with Applications, 100, 234–244. https://doi.org/10.1016/j.eswa.2018.01.037*
6. *Kadam, A. (2024). Human-in-the-loop dynamic graph learning for adaptive fraud detection. Journal of Artificial Intelligence and Big Data, 4(1), 12–28.*
7. *Kellogg, K. C., Valentine, M. A., & Christin, A. (2020). Algorithms at work: The new contours of algorithmic management. Academy of Management Annals, 14(1), 366–410. https://doi.org/10.5465/annals.2018.0174*
8. *Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. Human Factors, 46(1), 50–80. https://doi.org/10.1518/hfes.46.1.50_30392*
9. *Parent-Rocheleau, X., & Parker, S. K. (2022). Algorithms as work designers: How algorithmic management influences the design of jobs. Human Resource Management Review, 32(3), 100838. https://doi.org/10.1016/j.hrmr.2021.100838*
10. *Parveen, R. (2025). AN EXPLAINABLE AI FRAMEWORK FOR ETHICAL FRAUD PREVENTION IN U. S FEDERAL WELFARE PROGRAMS. International Journal of Applied Mathematics, 38(11s), 1425-1435.*
11. *Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., & Dean, J. (2017). Outrageously large neural networks: The sparsely-gated Mixture-of-Experts layer. arXiv preprint arXiv:1701.06538. https://doi.org/10.48550/arXiv.1701.06538*
12. *Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., & Christiano, P. F. (2020). Learning to summarize with human feedback. Advances in Neural Information Processing Systems (NeurIPS), 33, 3008–3021.*
13. *Vallarino, M. (2024). Hybrid Mixture-of-Experts architectures using LSTM and Transformer blocks for complex financial anomaly detection. IEEE Transactions on Neural Networks and Learning Systems, 35(8), 10412–10425.*
14. *Yang, L., Dong, X., & Wang, Y. (2024). Integrating dynamic Mixture-of-Experts with synthetic oversampling for imbalanced bank account fraud detection. Knowledge-Based Systems, 284, 111290. https://doi.org/10.1016/j.knosys.2023.111290*
15. *Yuksel, S. E., Wilson, J. N., & Gader, P. D. (2012). Twenty years of Mixture of Experts. IEEE Transactions on Neural Networks and Learning Systems, 23(8), 1177–1193. https://doi.org/10.1109/TNNLS.2012.2200299*
16. *Zhang, L. (2025). Governance and auditing frameworks for algorithmic management in human resources. Computers in Human Behavior, 150, 107982.*

17. *Chen, T., &Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794.* https://doi.org/10.1145/2939672.2939785

18. *Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems (NeurIPS), 30, 4765–4774.*

19. *Chawla, N. V., Bowyer, K. W., Hall, L. O., &Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357.* https://doi.org/10.1613/jair.953

20. *Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114.* https://doi.org/10.48550/arXiv.1312.6114

21. *Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., &Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS), 30, 5998–6008.*