Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Designing Robust AI Systems for Insurance Fraud Detection: Lessons from Real-World Deployments

Naresh Babu Goolla

IMR Soft LLC., USA

ARTICLE INFO

Received: 04 Nov 2025

Revised: 24 Dec 2025

Accepted: 03 Jan 2026

ABSTRACT

Insurance fraud detection technology has been continuously challenged by lingering issues of fraudulent schemes that continually evolve to evade traditional rule-based systems. Fraud rings, which are more sophisticated and coordinated criminal networks, find that manual investigation processes and static detection mechanisms are insufficient. Artificial intelligence technologies offer more advanced features through pattern recognition, behavioral analysis, and the automation of anomaly detection. Machine learning architectures employ both supervised and unsupervised learning approaches, depending on the availability of labeled training data and the specific types of fraud targeted. Supervised methods, including random forests, gradient boosting machines, and support vector machines, learn decision boundaries from historical patterns of fraud. Unsupervised techniques, such as isolation forests and autoencoders, identify anomalous claims that deviate from normal distributions without requiring explicit fraud labels. Feature engineering transforms raw claims data into meaningful patterns by incorporating temporal, behavioral, and network features, thereby capturing the relationships between entities. Successful production systems leverage ensemble methods combining multiple detection models to improve reliability. Technical failures primarily stem from model drift as fraud tactics evolve beyond the patterns in training data and extreme class imbalance, where legitimate claims vastly outnumber fraudulent submissions. The article examines architectural decisions, data pipeline configurations, and model management strategies, differentiating successful implementations from failed deployments. Understanding the technical factors that enable sustained performance proves essential for effective AI implementation in insurance fraud detection environments.

Keywords: Artificial Intelligence, Fraud Detection, Machine Learning, Anomaly Detection, Ensemble Methods, Model Drift

I. Introduction

The insurance industry, which includes property, casualty, health, and life insurance sectors, is the root cause of the fraud to which the most challenges are faced. The fraudulent submissions inflate the losses of the industry, which, in addition to the direct claim payouts, also include investigation costs, legal expenses, and increased premiums for policyholders [1]. Traditional fraud detection mechanisms relied heavily on manual review processes conducted by specialized investigation units. Static rule-based systems evaluated claims against predefined criteria established from historical fraud patterns [1]. Traditional methods have difficulties in detecting advanced fraud schemes that continuously change in order to bypass their detection mechanisms. Insurance fraud is a phenomenon that occurs in different kinds of hard fraud, soft fraud, and opportunistic fraud. Hard fraud involves deliberate staging of losses or fabrication of claims with the intent to defraud insurers. Soft fraud occurs when claimants exaggerate legitimate claims or misrepresent circumstances to increase

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

settlement amounts [1]. As a result of the activities of the organized fraud rings, many actors are involved in the execution of complex schemes that involve accidents that are staged accidents, inflated medical treatments that are inflated, or fictitious property damage that is fictitious. Manual investigation methods are not capable of detecting the multifaceted fraud tactics that confront these methods, which happen to span multiple claims and also involve coordinated criminal networks.

Artificial intelligence technologies provide better potential for fraud detection through pattern recognition, behavioral analysis, and the automated identification of anomalies. Machine learning algorithms examine enormous datasets so they can detect faint patterns that human investigators or rule-based systems are not able to see [2]. Besides, the change from using the conventional detection methods to AI-driven systems is not an easy one, as it is fraught with a lot of technical challenges. Interpretability of the model has to do with compliance with regulations and acceptance by the investigators. Computational resource requirements increase substantially when processing high-volume claim submissions in real-time. Integration complexity with legacy claims management systems creates implementation barriers [2].

Current literature predominantly examines algorithm performance metrics during controlled testing phases. Research focuses on classification accuracy, precision, and recall measurements without addressing operational constraints encountered in production environments [3]. Data quality issues significantly impact model performance yet receive limited attention in academic studies. System integration challenges that affect real-world effectiveness remain underexplored. The gap between theoretical algorithm performance and practical deployment outcomes requires comprehensive analysis [3].

Production environments present challenges distinct from experimental settings. Model drift degrades performance as fraud tactics evolve beyond patterns present in training data. Class imbalance between fraudulent and legitimate claims creates training difficulties for supervised learning approaches. False positive rates impact operational efficiency by requiring human review of incorrectly flagged legitimate claims [3]. Processing latency constraints demand efficient algorithms capable of real-time decision-making without compromising detection accuracy.

This paper examines architectural decisions, data pipeline configurations, and model management strategies that differentiate successful AI fraud detection implementations from failed deployments. System design choices impact detection accuracy, processing latency, and operational sustainability. Understanding technical factors that enable sustained performance beyond initial deployment phases proves essential for effective AI implementation in insurance fraud detection.

Table 1. Comparative Analysis of Fraud Detection Methodologies [1, 2, 3].

Methodology Category	Techniques Employed	Primary Limitation
Rule-Based Systems	Static criteria evaluation, Expert system logic	Inadequate against novel fraud schemes
Statistical Analysis	Outlier detection, Distribution analysis	Limited pattern recognition capabilities
Supervised Learning	Decision trees, Neural networks, Support vector machines	Requires extensive labeled training data
Unsupervised Learning	Clustering, Anomaly detection, Autoencoders	Higher false positive rates
Deep Learning	Convolutional neural networks, Recurrent neural networks, Graph	Computational complexity and interpretability

Research Article

	neural networks	challenges
Hybrid Architectures	Ensemble methods, Multi-model combinations	Integration complexity with legacy systems

II. Related Work and Methodology

Related Work

Insurance fraud detection has attracted substantial attention from machine learning research communities over the past decade. Early approaches relied predominantly on statistical analysis techniques, identifying outliers in claims distributions. Rule-based expert systems encoded domain knowledge from experienced fraud investigators into decision trees and logic rules. These traditional methods demonstrated effectiveness against known fraud patterns but struggled with novel schemes lacking predefined detection criteria.

Recent literature explores supervised machine learning applications for fraud classification. Decision trees, neural networks, and support vector machines are subjected to thorough scrutiny in various fields such as credit card fraud, insurance claims fraud, and financial transaction monitoring. Ensemble learning strategies that combine several base classifiers have better performance than individual algorithms. Random forests and gradient boosting machines appear frequently in published fraud detection frameworks, demonstrating robust accuracy across diverse datasets.

Unsupervised learning techniques address the limitations of supervised approaches, requiring extensive labeled training data. Clustering algorithms group similar transactions, identifying outlier clusters potentially representing fraudulent activity. Autoencoders learn compressed representations of normal transaction patterns, flagging instances with high reconstruction errors as anomalies. Isolation forests partition feature spaces, identifying instances requiring fewer splits for isolation as potential fraud cases.

Deep learning architectures introduce advanced capabilities for processing complex unstructured data. Convolutional neural networks analyze image data from claims documentation, detecting manipulated photographs or fabricated damage evidence. Recurrent neural networks process sequential transaction histories identifying temporal patterns indicative of coordinated fraud schemes. Graph neural networks examine relationship networks between claimants, providers, and beneficiaries uncovering organized fraud rings.

Existing literature predominantly evaluates algorithm performance using historical datasets with known fraud labels. Published studies report accuracy, precision, recall, and F1-scores comparing different algorithms under controlled experimental conditions. However, significant gaps exist regarding operational deployment challenges encountered in production environments. Limited research examines model degradation over time as fraud tactics evolve. Class imbalance handling receives theoretical treatment, but practical implications for false positive management remain underexplored. Integration complexity with legacy systems and computational constraints for real-time processing lacks comprehensive documentation.

Methodology

The article employs a comparative analytical framework examining successful and failed AI fraud detection implementations. The methodology synthesizes technical architectures, deployment strategies, and operational outcomes across multiple insurance fraud detection systems. Analysis focuses on identifying architectural decisions and system design choices, differentiating sustainable implementations from failed deployments.

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Technical architecture analysis examines machine learning model selection criteria, evaluating supervised versus unsupervised learning trade-offs. Feature engineering pipelines receive detailed examination documenting temporal, behavioral, and network feature extraction techniques. Data preprocessing workflows, including missing value handling, feature scaling, and categorical encoding methods, undergo systematic evaluation.

Successful implementation patterns receive characterization through anomaly detection approaches and ensemble method configurations. Analysis documents specific algorithm combinations, threshold selection strategies, and model updating protocols enabling sustained detection performance. Failed implementation analysis identifies common failure modes, including inadequate drift detection mechanisms, insufficient retraining protocols, and poor false positive management strategies.

Model drift analysis examines temporal performance degradation patterns across different fraud detection systems. The methodology categorizes drift types, including sudden, gradual, incremental, and recurring drift patterns. Adaptation strategies receive evaluation based on effectiveness, computational efficiency, and implementation complexity. Class imbalance handling techniques undergo comparative assessment examining oversampling, undersampling, and cost-sensitive learning approaches.

The analytical framework contributes to systematic documentation of technical constraints affecting real-world fraud detection effectiveness. Novel contributions include identifying hybrid architecture requirements combining multiple detection strategies, characterizing drift adaptation protocols essential for sustained performance, and documenting false positive management approaches balancing detection sensitivity against operational capacity. The methodology provides practitioners with actionable insights for designing robust fraud detection systems, avoiding common implementation pitfalls documented through failed deployment analysis.

III. Technical Architecture of AI Fraud Detection Systems

A. Machine Learning Model Selection

AI-based fraud detection systems employ diverse machine learning architectures depending on specific fraud types and available training data characteristics. Supervised learning models require labeled historical data identifying confirmed fraudulent and legitimate claims. AdaBoost algorithms construct ensemble classifiers by iteratively training weak learners on weighted training samples [4]. Each iteration increases weights for misclassified examples, forcing subsequent learners to focus on difficult cases. Majority voting mechanisms combine predictions from multiple base classifiers to reach final decisions [4].

Random forests build multiple decision trees using bootstrap sampling techniques. Each tree receives random subsets of features during node splitting processes. Gradient boosting machines create sequential models where each new model corrects residual errors from previous predictions. Support vector machines identify optimal separating hyperplanes in high-dimensional feature spaces [4]. Algorithm selection depends on dataset characteristics, computational constraints, and interpretability requirements for regulatory compliance.

Unsupervised learning approaches address limitations of supervised methods when labeled fraud examples remain scarce. Isolation forests operate through recursive random partitioning of feature spaces [5]. The algorithm constructs isolation trees by randomly selecting features and split values. Anomalous instances require fewer partitions for isolation compared to normal instances clustered in

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

dense regions. Path lengths from root nodes to terminal leaves provide anomaly scores for individual claims [5].

Extended isolation forest variants address limitations in original implementations. Standard isolation forests struggle with local anomalies embedded within normal data clusters. Enhanced versions incorporate branching improvements and scoring adjustments to detect localized anomalies more effectively [5]. Autoencoders learn compressed representations of legitimate claim patterns through encoder-decoder neural network architectures. Claims exhibiting high reconstruction errors indicate deviations from learned normal distributions.

B. Feature Engineering and Data Processing

Feature engineering transforms raw claims data into meaningful patterns analyzable by machine learning models. Temporal features capture submission timing patterns, policy duration before claims, and seasonal variations in claim frequencies [6]. Amount-based features include claim values, comparative ratios against policy limits, and statistical deviations from typical claim amounts for specific coverage types. Transaction frequency features track claim submission rates per policyholder over defined time windows [6].

Behavioral features analyze interaction patterns throughout claims processing lifecycles. Communication frequency with adjusters, document submission completeness, and response latency to information requests provide behavioral indicators. Categorical features encode claim types, coverage categories, geographic locations, and involved party relationships [6]. Binary encoding represents the presence or absence of specific claim attributes such as attorney involvement or witness availability.

Network features represent relationships between entities participating in claims ecosystems. Graph construction connects claimants, service providers, witnesses, and beneficiaries through shared claim interactions [6]. Centrality measures identify highly connected nodes, potentially representing fraud ring coordinators. Community detection algorithms reveal tightly connected subgraphs, suggesting organized fraud networks. Feature scaling normalizes numerical attributes to comparable ranges, preventing dominance by large-magnitude features. Missing value imputation addresses incomplete data through statistical methods or learned imputation models [6]. External data integration validates claims against weather databases, medical procedure standards, and repair cost benchmarks.

Table 2. Machine Learning Algorithms and Feature Types in Fraud Detection Systems [4, 5, 6].

Algorithm Category	Specific Techniques	Feature Processing Approach
Supervised Learning	Random Forests, Gradient Boosting Machines, Support Vector Machines, AdaBoost	Requires labeled historical data with confirmed fraud cases
Unsupervised Learning	Isolation Forests, Autoencoders, Clustering Algorithms	Identifies anomalies without explicit fraud labels
Temporal Features	Claim submission timing, Policy inception comparison, Seasonal pattern analysis	Captures timing patterns and temporal correlations
Behavioral Features	Communication frequency, Document submission sequences, Response time	Analyzes claimant interaction

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

	analysis	patterns
Network Features	Graph neural networks, Entity relationship mapping, Fraud ring detection	Represents connections between claimants, providers, and witnesses

IV. Successful Implementation Patterns

A. Anomaly Detection in Claims Patterns

Successful AI implementations leverage anomaly detection algorithms to identify claims exhibiting unusual characteristics compared to historical norms. Novel fraud patterns emerge continuously as fraudsters adapt tactics to circumvent existing detection mechanisms. Novelty detection approaches identify previously unseen patterns deviating from established normal behavior baselines [7]. The challenge involves distinguishing genuine innovations in fraud tactics from legitimate, unusual claims requiring different processing approaches.

Detection systems analyze multidimensional feature spaces encompassing claim amounts, timing patterns, provider relationships, and documentation completeness. Novelty assessment evaluates claims against learned representations of normal claim distributions [7]. Reference-based methods compare new claims to established knowledge bases of legitimate claim characteristics. Temporal analysis tracks the evolution of claim patterns over time, identifying emerging fraud schemes before widespread adoption.

Isolation forest algorithms partition feature spaces through recursive random splitting. Anomalous claims require fewer partitions for isolation compared to normal claims concentrated in dense feature space regions. Path lengths from root nodes to terminal leaves provide anomaly scores quantifying deviation magnitudes [7]. Shorter paths indicate stronger anomalies warranting detailed investigation.

Autoencoder neural networks learn compressed representations of legitimate claim patterns through encoder-decoder architectures. Encoders transform input claims into lower-dimensional latent representations capturing essential normal claim characteristics. Decoders reconstruct original claims from compressed representations [7]. Reconstruction errors quantify deviations from learned normal patterns. High reconstruction errors flag claims exhibiting significant differences from training data distributions. The approach detects novel fraud schemes absent from training datasets by identifying normality deviations rather than matching known fraud signatures.

B. Ensemble Methods for Robust Detection

Successful production systems employ ensemble approaches combining multiple detection models to improve reliability and reduce false positives. XGBoost implements scalable gradient boosting through optimized tree construction algorithms [8]. The system builds ensemble models by sequentially adding trees, minimizing objective functions that combine training loss and model complexity. Regularization terms prevent overfitting by penalizing complex tree structures with excessive leaf nodes or large leaf weights [8].

Gradient boosting machines train decision trees sequentially, with each tree correcting residual errors from previous iterations. Second-order gradient information guides tree construction, enabling more accurate approximation of objective functions [8]. Column block storage structures facilitate parallel tree learning by sorting features once and reusing sorted orders across iterations. Cache-aware access patterns optimize memory bandwidth utilization during gradient statistics accumulation [8].

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

Sparsity-aware algorithms handle missing values natively without requiring imputation preprocessing. Default directions assigned to tree nodes route instances with missing feature values efficiently [8]. Weighted quantile sketch algorithms enable approximate tree learning with instance weights. The approach selects candidate split points representing weighted quantile distributions of training data [8].

Stacking ensembles combine predictions from diverse base models, including logistic regression, decision trees, and neural networks. Base models train independently, generating prediction outputs for meta-model training. Meta-models learn optimal combination strategies, weighting individual base model contributions. The architecture leverages complementary algorithm strengths with tree-based models capturing complex interactions while linear models provide stable baseline predictions [8].

Table 3. Successful AI Implementation Components and Detection Mechanisms [7, 8].

Implementation Component	Technical Approach	Key Characteristics
Anomaly Detection	Isolation forests with path length scoring	Partitions feature space identifying unusual claims
Autoencoder Networks	Encoder-decoder architecture with reconstruction error analysis	Learns compressed representations of legitimate patterns
XGBoost Implementation	Sequential tree training with gradient optimization	Regularization prevents overfitting through complexity penalties
Ensemble Stacking	Combines logistic regression, decision trees, and neural networks	Meta-models weight individual base model predictions
Sparsity Handling	Native missing value processing with default directions	Handles incomplete data without imputation preprocessing

V. Technical Failures and Limitations

A. Model Drift and Adaptation Challenges

AI fraud detection systems frequently fail due to model drift where performance degrades over time as fraud tactics evolve beyond patterns present in training data. Concept drift describes scenarios where relationships between input features and target variables change over time [9]. Sudden drift occurs when abrupt shifts in fraud patterns emerge instantaneously. Gradual drift manifests through slow progressive changes in fraudulent behavior characteristics. Incremental drift involves step-wise transitions between different fraud pattern regimes. Recurring drift patterns cycle between previously observed fraud tactics reappearing after periods of dormancy [9].

Fraudsters continuously adapt techniques to circumvent detection systems. Static models become progressively less effective against evolving tactics. Detection mechanisms must identify drift occurrence timing and characterize drift severity. Statistical process control methods monitor prediction error rates, detecting significant deviations from expected performance levels [9]. Hypothesis testing approaches compare current data distributions against historical baseline distributions. Sequential analysis techniques enable early drift detection before substantial performance degradation occurs.

Adaptation strategies determine system responses after drift detection. Model retraining uses new labeled data, rebuilding classifiers from scratch. Ensemble updating maintains multiple models

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://jisem-journal.com/

Research Article

representing different concept periods, weighting models based on current relevance [9]. Windowing techniques restrict training data to recent observations, assuming temporal proximity indicates concept similarity. Seasonal variations in claim patterns create cyclical distribution shifts. Regulatory changes modify claim processing requirements, altering feature distributions. Macroeconomic factors influence claim frequencies and characteristics [9].

Failed implementations lack automated drift detection mechanisms and established retraining protocols. Manual monitoring proves insufficient for identifying gradual drift patterns. Computational costs of continuous retraining constrain practical deployment options. Memory requirements for storing historical data across multiple concept periods create resource management challenges [9].

B. Imbalanced Data and False Positive Management

Extreme class imbalance in fraud detection creates significant technical challenges where legitimate claims vastly outnumber fraudulent submissions. Standard accuracy metrics mislead when fraudulent claims constitute minority classes in training datasets [10]. Evaluation complexity increases when imbalance ratios exceed typical thresholds. Within-class imbalance occurs when fraud subcategories exhibit different representation levels. Small disjuncts represent sparse regions in feature space where limited training examples provide insufficient learning signals [10].

Data-level solutions modify training set compositions through resampling techniques. Random undersampling reduces majority class instances, risking information loss from discarded legitimate claims. Informed undersampling selects representative majority class subsets preserving important decision boundary examples [10]. Synthetic oversampling generates artificial minority class instances through interpolation mechanisms. Borderline-SMOTE focuses synthetic generation on minority class examples near decision boundaries. ADASYN adapts synthetic sample generation rates based on local class distributions [10].

Algorithm-level approaches modify learning algorithms, accommodating imbalanced distributions. Cost-sensitive learning assigns asymmetric misclassification penalties, penalizing minority class errors more heavily. One-class learning trains exclusively on majority class examples treating minority detection as novelty identification [10]. Ensemble methods combine multiple base classifiers trained on balanced data subsets. High false positive rates create operational burdens requiring human investigators to review numerous legitimate claims incorrectly flagged as suspicious. Investigation capacity constraints limit reviewable claim volumes. Excessive false positives degrade customer experience through unnecessary delays and intrusive verification requests [10].

Table 4. Technical Failure Modes and Mitigation Challenges [9, 10].

Failure Category	Manifestation	Technical Challenge
Sudden Drift	Abrupt shifts in fraud patterns	Requires immediate model adaptation
Gradual Drift	Progressive changes in fraudulent behavior	Difficult to detect before performance degradation
Incremental Drift	Step-wise transitions between fraud regimes	Demands continuous monitoring mechanisms
Recurring Drift	Cyclical reappearance of previous fraud tactics	Needs historical pattern recognition
Class Imbalance	Legitimate claims vastly outnumber fraudulent submissions	Standard accuracy metrics become misleading
False Positive Burden	Legitimate claims incorrectly flagged as suspicious	Creates operational strain on investigation teams

Conclusion

Implementing artificial intelligence systems for insurance fraud detection demonstrates considerable potential alongside significant technical challenges. Successful deployments share common characteristics, including hybrid architectures combining multiple detection strategies, continuous model monitoring infrastructure, and sophisticated feature engineering pipelines capturing temporal, behavioral, and network patterns. Production systems require ensemble approaches rather than single-model solutions, recognizing that diverse fraud tactics demand complementary detection mechanisms working cooperatively. Failed implementations suffer from inadequate handling of concept drift as fraudulent behaviors evolve beyond training data representations. Class imbalance between legitimate and fraudulent claims creates persistent training difficulties for supervised learning algorithms. False positive rates impact operational efficiency by requiring human investigators to review incorrectly flagged legitimate claims. Processing latency constraints demand efficient algorithms capable of real-time decision-making without compromising detection accuracy.

Future development directions should prioritize continual learning architectures that adapt to evolving fraud patterns without complete model retraining cycles. Explainable artificial intelligence techniques, providing investigators with actionable insights rather than opaque predictions, enhance system utility and regulatory compliance. Federated learning approaches enable collaborative fraud detection across organizations while preserving sensitive data privacy requirements. Integrating large language models for analyzing unstructured claims documentation, including textual descriptions, medical reports, and witness statements, represents promising enhancement opportunities. Graph neural networks analyzing entity relationships across claims ecosystems identify coordinated fraud rings operating systematically. Automated drift detection strategies paired with adaptive retraining procedures keep account models up to date as fraud techniques change. Enterprises need to allocate a high amount of resources to the setup of the monitoring system, data quality management, and the establishment of model governance structures to reap long-term benefits from AI-powered fraud detection. On the one hand, technical architectures should optimize detection performance; on the other hand, they should consider operational feasibility so that system results can interact smoothly with the manual investigation process. Machine learning systems require ongoing maintenance and adaptation rather than static deployment approaches, treating models as fixed solutions.

References

[1] Richard A. Derrig, "INSURANCE FRAUD," The Journal of Risk and Insurance, 2002. [Online]. Available: https://web.actuaries.ie/sites/default/files/erm-resources/insurance_fraud.pdf

[2] Sharifur Rahman et al., "An Exploration of Artificial Intelligence Techniques for Optimizing Tax Compliance, Fraud Detection, and Revenue Collection in Modern Tax Administrations," TensorGate, 2024. [Online]. Available: https://www.researchgate.net/profile/Rafiqus-Khan/publication/387585676

[3] Lutfun Nahar Lata et al., "A Comprehensive Survey of Fraud Detection Techniques," International Journal of Applied Information Systems, 2015. [Online]. Available: https://www.ijais.org/research/volume10/number2/lata-2015-ijais-451471.pdf

[4] KULDEEP RANDHAWA et al., "Credit Card Fraud Detection Using AdaBoost and Majority Voting," IEEE Access, 2018. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8292883

Research Article

[5] YOUSRA CHABCHOUB et al., "An In-Depth Study and Improvement of Isolation Forest," IEEE access, 2021. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9684896

[6] Pooja Tiwari et al., "CREDIT CARD FRAUD DETECTION USING MACHINE LEARNING: A STUDY," arXiv, 2021. [Online]. Available: https://arxiv.org/pdf/2108.10005

[7] Yi Zhao and Chengzhi Zhang, "A Review on the Novelty Measurements of Academic Papers," arXiv, 2025. [Online]. Available: https://arxiv.org/pdf/2501.17456?

[8] Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," ACM, 2016. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/2939672.2939785

[9] J. Lu et al., "Learning under Concept Drift: A Review," arXiv, 2020. [Online]. Available: https://arxiv.org/pdf/2004.05785

[10] Bartosz Krawczyk, "Learning from imbalanced data: open challenges and future directions," Springer Nature, 2016. [Online]. Available: https://link.springer.com/article/10.1007/s13748-016-0094-0?TB_iframe=true&error=cookies_not_supported&code=a3e33168-782e-41e5-8585-e731754069d2