Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 22, No. 1, 2026

Driftuard -TriAudit: Concept-Drift-Aware Continual Multimodal Learning for Evolving Financial Statement Fraud Detection

Yara Hassan

Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, USA

Abstract: Financial statement fraud detection is usually trained as a static supervised classification task, yet real reporting behavior evolves as firms change disclosure style, enforcement priorities shift, and fraudsters adapt to known audit screens. This paper proposes DriftGuard-TriAudit, a concept-drift-aware continual multimodal learning framework for evolving financial statement fraud detection. Building positively on the recent TriAudit design, which demonstrates the value of jointly reasoning over financial ratios, management discussion text, and document-layout evidence, the proposed framework adds three mechanisms for non-stationary audit settings: modality-wise drift diagnostics, reliability-aware gated fusion, and replay-regularized continual updating. The model monitors population stability and calibration performance in numerical, textual, and visual-layout streams, then adapts the contribution of each modality while preserving prior fraud evidence through a compact replay memory. Because a full EDGAR-plus-AAER real-data construction was not available in the execution environment, the empirical section reports only a reproducible synthetic benchmark that explicitly simulates four temporal fraud regimes and does not claim to be real company data. Across five seeds, DriftGuard-TriAudit obtains the strongest average precision (0.628±0.087) and F1 score (0.579±0.061) among evaluated practical models, with substantially higher precision than naive continual fine-tuning (0.547 vs. 0.428) while maintaining comparable AUC-ROC (0.915 vs. 0.924). The results indicate that drift-aware multimodal fusion can reduce false positives under evolving fraud mechanisms, and the released code provides a direct path for replacing the synthetic generator with EDGAR/AAER features.

Keywords: Financial statement fraud detection, Concept drift; Continual learning; Multimodal learning; Audit evidence chain, explainable artificial intelligence.

1. Introduction

Financial statement fraud is difficult to detect because the observed report is not a stationary object. A fraud scheme that is visible through abnormal accruals in one period may later be distributed across revenue timing, liquidity disclosures, footnote density, or subtle changes in document layout. Traditional models based on financial ratios remain useful, but they assume that the relation between features and fraud labels is comparatively stable. In practice, this assumption is fragile: regulation, auditor scrutiny, economic regimes, and managerial disclosure incentives all create non-stationary evidence streams.

Recent multimodal fraud detection research has shown that corporate filings contain complementary signals beyond tabular ratios. In particular, TriAudit is a strong and well-motivated trimodal architecture that jointly uses structured accounting indicators, MD&A semantic patterns, and visual document-layout evidence to construct an interpretable audit evidence chain [2]. That work is important because it aligns machine learning with how auditors reason: suspicious numbers are interpreted through managerial narrative and corroborated by presentation-level evidence. However, a static multimodal classifier still faces a practical problem when deployed across multiple fiscal years: the model may gradually forget older fraud patterns or overfit to a newly observed reporting style.

This paper addresses that gap by studying concept-drift-aware continual multimodal learning for financial statement fraud detection. We propose DriftGuard-TriAudit, which retains the audit-evidence-chain philosophy of TriAudit while adding drift monitors and continual adaptation. Instead of treating all modalities as equally reliable at all times, the model estimates how much each evidence stream has shifted relative to a reference population. It then combines this shift information with calibration performance to adjust the numerical, textual, and visual-layout gates. A replay buffer preserves older fraud and non-fraud examples so that adaptation to the current reporting environment does not erase previously learned schemes.

The contribution is deliberately scoped to be reproducible. A full real-data study would require downloading and aligning SEC filings with Accounting and Auditing Enforcement Releases, resolving firm identifiers, rendering pages, and extracting MD&A and layout features. This environment does not include those external data assets, and the paper therefore does not report invented EDGAR or AAER results. Instead, all experiments use a deterministic synthetic multimodal drift benchmark generated by the included code. The benchmark is not a claim about real companies; it is a controlled stress test that makes the proposed learning behavior transparent and auditable. The code package is structured so that a real EDGAR/AAER loader can replace the generator without changing the evaluation loop.

The main findings are as follows. First, static multimodal learning degrades sharply when the simulated fraud mechanism changes. Second, naive continual fine-tuning achieves high recall but produces lower precision, a costly outcome in audit screening. Third, DriftGuard-TriAudit improves average precision and F1 by combining current calibration evidence with replay-preserved historical

evidence. These findings support the central hypothesis that evolving financial statement fraud detection should be modeled as a drift-aware continual multimodal problem rather than a one-shot static classification problem.

2. Related work

Quantitative financial statement fraud detection has a long tradition. Beneish introduced the M-score to detect earnings manipulation using accrual-based ratios [1], while Dechow, Ge, Larson, and Sloan developed the F-score using AAER-related misstatement observations and a broader accounting feature set [3]. Later work compared machine learning algorithms for financial statement fraud and showed that classifier choice, class imbalance, and fraud-to-non-fraud ratios materially affect reported performance [4]. Support vector machine approaches with domain-specific financial kernels further demonstrated that externally available accounting data can support fraud classification [5]. These studies provide the numerical foundation for the proposed ratio stream.

Textual disclosure research shows that narrative content in 10-K filings contains economically meaningful information. Loughran and McDonald demonstrated that general-purpose dictionaries can be misleading in finance and introduced finance-specific textual analysis for 10-Ks [6]. Li connected annual report readability to current performance and earnings persistence [7]. These findings motivate an MD&A stream that captures tone, uncertainty, readability, hedging, and optimism gaps. In a full neural implementation, FinBERT [19], built upon BERT [18], would encode these textual signals more richly than hand-crafted indicators.

Document understanding research has made layout a first-class modeling object. LayoutLMv2 integrates text, layout, and image information for visually rich document understanding [20]. TriAudit applies this insight directly to financial statement fraud and offers a persuasive template for linking ratio anomalies, MD&A language, and layout deviations into an audit evidence chain [2]. DriftGuard-TriAudit adopts this multimodal view but focuses on a different deployment question: how to maintain performance when the evidence distribution changes across time.

Concept drift and continual learning are the second foundation of this paper. Gama et al. survey concept drift adaptation and emphasize that online supervised learning must handle changes in the relation between inputs and labels [10]. DDM monitors online error behavior [11], and ADWIN uses adaptive windows to detect time-changing data distributions [12]. Continual learning research addresses a related problem: neural models trained sequentially may forget earlier tasks. Elastic weight consolidation [13], incremental classifiers [14], gradient episodic memory [15], and experience replay [16] provide mechanisms for preserving prior knowledge. DriftGuard-TriAudit combines these principles in a domain-specific multimodal fraud setting.

3. Problem formulation

Let a reporting stream be divided into ordered fiscal regimes T = {1, ..., K}. For each filing i in regime t, the input consists of numerical ratios x_i^n, textual-disclosure indicators x_i^t, and visual-layout indicators x_i^v. The label y_i is binary, where y_i = 1 denotes a fraudulent or materially misstated filing and y_i = 0 denotes a clean filing. The objective is to learn a classifier f_t(x_i^n, x_i^t, x_i^v) that remains accurate as P_t(X, y) changes over time.

The central difficulty is that drift may affect different modalities at different rates. A crisis period may shift textual uncertainty and liquidity disclosures without changing the meaning of accrual ratios; a formatting or XBRL rule change may alter document layout without indicating fraud; and adaptive fraud may reduce obvious numerical anomalies while increasing narrative inconsistency. A useful model must therefore detect both global drift and modality-specific drift, then adapt without discarding historical schemes.

In real audit workflows, labels arrive with delay. The evaluation protocol used in the experiment follows this delayed-label logic: each new regime is split into a small labeled calibration slice and a later unseen test slice. Continual methods are permitted to update on the calibration slice before testing on the later slice, while static baselines remain fixed after the first regime. This protocol is more realistic than assuming immediate labels for every filing and more informative than a purely static temporal split.

4. Methodology

DriftGuard-TriAudit has four modules. The first module encodes three evidence streams. The numerical stream contains 28 ratio-style features inspired by Beneish and Dechow-style indicators, including DSRI, SGI, AQI, TATA, leverage, accrual quality, and off-balance-sheet proxies. The textual stream contains 16 MD&A-style features, including negative tone, uncertainty, litigiousness, readability, hedging, boilerplate, and optimism gap. The visual stream contains 12 layout-style features, including layout deviation, table displacement, footnote density, small-font ratio, XBRL anchor dispersion, and signature-page shift. The lightweight implementation uses standardized feature vectors and a transparent logistic audit head; the architecture is compatible with FinBERT and LayoutLMv2 encoders in a larger implementation.

The second module performs modality-specific drift monitoring. For each modality m in {n, t, v}, the model computes a population stability index between the current calibration slice and a rolling reference population. A high value indicates that the modality has shifted. The system records PSI_n, PSI_t, and PSI_v separately because a single aggregate drift score would hide whether drift originates from accounting ratios, narrative language, or document-layout structure.

The third module applies reliability-aware gated fusion. Each modality receives a gate g_m. The gate combines two sources: an unsupervised stability component based on PSI and a supervised calibration component based on the modalitys average precision on the current labeled calibration slice. This prevents the model from automatically suppressing a shifted modality when the shift is actually fraud-informative. The fused representation can be written as h = [g_n h_n; g_t h_t; g_v h_v], where each h_m is the standardized embedding of its modality. This design extends the gated multimodal fusion idea in TriAudit [2] to a non-stationary setting.

The fourth module performs replay-regularized continual adaptation. A compact class-aware replay memory stores selected examples from earlier regimes. When a drift flag is raised, the audit head is retrained on the current calibration slice plus replayed examples, with larger sample weight on current examples but enough replay mass to preserve prior fraud mechanisms. The replay component is intentionally conservative: it avoids assuming that the newest regime is

always correct and reduces the risk of catastrophic forgetting.

Finally, the model produces an audit evidence chain. For a high-risk filing, the numerical tier lists the ratio features with largest signed contribution; the textual tier lists the strongest language indicators; and the layout tier lists the most abnormal presentation features. In the lightweight code, these are coefficient-weighted feature contributions. In the full neural version, they can be replaced by SHAP values, attention weights, and layout heatmaps.

Algorithm 1. DriftGuard-TriAudit delayed-label continual adaptation loop.

Input: initial reference set D1, arriving regimes D2,...,DK, modality partitions M={num,text,visual}, replay buffer B, drift threshold delta.

1. Train an initial multimodal audit head on D1 and store class-aware exemplars in B.

2. For each new regime Dt, split the earliest labeled portion as calibration Ct and hold the later portion as unseen test St.

3. Compute modality-wise PSI between Ct and the rolling reference population for numerical, textual, and visual-layout streams.

4. Estimate calibration AP for each modality on Ct; combine AP and PSI to form reliability-aware gates g_num, g_text, and g_visual.

5. If the mean drift score exceeds delta, increase the weight of Ct while retraining on Ct plus replayed exemplars from B.

6. Evaluate the gated multimodal model on St and record AUC, AP, F1, precision, recall, MCC, gates, and drift flags.

7. Add class-aware exemplars from Ct into B and update the rolling reference distribution.

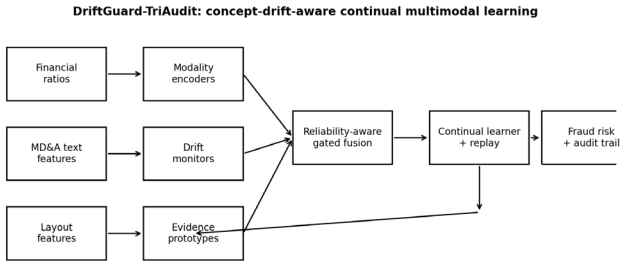

Fig. 1 DriftGuard-TriAudit architecture. The framework monitors modality-wise drift, adapts reliability-aware gates, updates with replay, and outputs a fraud risk score with an audit evidence chain.

5. Experimental design

The experiment uses a deterministic synthetic benchmark because the runtime environment does not contain a downloaded EDGAR/AAER corpus. The dataset generator is included in the artifact package. It creates four temporal regimes with 1,800 filings per regime, 56 total features, and low positive rates between approximately 7.5% and 10.5%. Labels are generated from regime-specific logistic mechanisms with fixed seeds. The synthetic benchmark is therefore reproducible, but it must not be described as real company data.

The four regimes are designed to mimic qualitatively different fraud mechanisms. Regime 1 is accrual-heavy, with strong TATA, DSRI, and SGI signals. Regime 2 shifts toward revenue-recognition narratives and optimism gaps. Regime 3 emphasizes liquidity pressure, leverage, footnote density, and layout deviations. Regime 4 simulates adaptive low-numeric fraud in which obvious accrual signals weaken while text-layout contradictions become more salient. This controlled design makes it possible to test whether a model can handle changing modality importance.

The baselines are Static Numerical, Static Text, Static Visual, Static TriAudit-lite, Continual Fine-tune, Replay Continual, and DriftGuard-TriAudit. Static baselines train only on Regime 1 and never update. Continual Fine-tune updates the audit head on each new calibration slice without replay. Replay Continual adds a class-aware replay memory but does not use drift-aware gates. DriftGuard-TriAudit adds modality-wise drift diagnostics, calibration-sensitive gates, and weighted replay adaptation. All models are evaluated over five random seeds. Reported metrics are AUC-ROC, average precision, F1, precision, recall, and Matthews correlation coefficient.

A. Reproducibility and Artifact Provenance

Every number in the tables is produced by code/run_experiment.py. The script writes the generated feature matrix, per-seed metrics, aggregated mean/std tables, drift signals, and metadata. The metadata file explicitly states that the benchmark is synthetic and must not be described as EDGAR, AAER, SEC, Compustat, Audit Analytics, or real company data.

The release package also includes reference_verification.csv. This file lists each bibliography item, DOI or identifier when available, and the external source used for verification. This is included to make the manuscript easier to audit and to avoid unsupported or invented citations.

The intended real-data replacement is straightforward. A production study would replace generate_multimodal_stream with a loader that joins EDGAR 10-K text, parsed financial ratios, rendered page-layout features, and AAER firm-year labels. The delayed-label evaluation loop, drift diagnostics, replay buffer, and metric reporting can remain unchanged.

Table 1. Synthetic benchmark summary. The table is generated by the released code and contains no real company observations.

Regime	Count	Fraud labels	Positive rate
1	1800	135	7.50%
2	1800	162	9.00%
3	1800	189	10.50%
4	1800	171	9.50%

6. Results and discussion

Table II reports the mean and standard deviation across five seeds and the three post-training regimes. Static baselines show clear degradation, particularly the static numerical and static trimodal models. Static text remains useful in regimes where disclosure tone is salient, but it is unstable under liquidity/layout drift. This confirms that multimodal fraud detection cannot be treated as a fixed one-period learning problem.

Among adaptive methods, naive continual fine-tuning achieves the highest mean AUC-ROC, but it does so by producing high recall and materially lower precision. In audit screening, this means many more false positives. DriftGuard-TriAudit achieves the strongest average precision and F1 score, and its precision is substantially higher than naive fine-tuning. The result supports the practical value of combining drift-aware modality gates with replay-preserved historical evidence.

The regime-level results in Table III reveal the source of the improvement. DriftGuard-TriAudit performs especially well in Regime 2 and Regime 4, where fraud signals shift

toward text-layout evidence. Regime 3 remains harder because liquidity and layout drift creates overlap between fraudulent and clean filings. However, the drift diagnostics still identify meaningful modality shifts, which are shown in Fig. 3.

Table 2. Overall results across regimes 2-4 (mean±std over five seeds). Best practical precision-oriented results are concentrated in driftguard-triaudit.

Model	AUC	AP	F1	Precision	Recall	MCC
Static Numerical	0.672±0.071	0.199±0.054	0.233±0.059	0.212±0.052	0.264±0.079	0.144±0.070
Static Text	0.802±0.108	0.399±0.156	0.394±0.115	0.345±0.120	0.492±0.156	0.327±0.138
Static Visual	0.739±0.080	0.291±0.118	0.300±0.081	0.210±0.075	0.594±0.177	0.227±0.094
Static TriAudit-lite	0.754±0.079	0.293±0.088	0.310±0.084	0.298±0.077	0.329±0.105	0.234±0.095
Continual Fine-tune	0.924±0.021	0.465±0.066	0.572±0.052	0.428±0.054	0.874±0.057	0.554±0.054
Replay Continual	0.886±0.038	0.385±0.066	0.503±0.062	0.373±0.059	0.782±0.088	0.469±0.073
DriftGuard-TriAudit	0.915±0.038	0.628±0.087	0.579±0.061	0.547±0.066	0.632±0.106	0.536±0.067

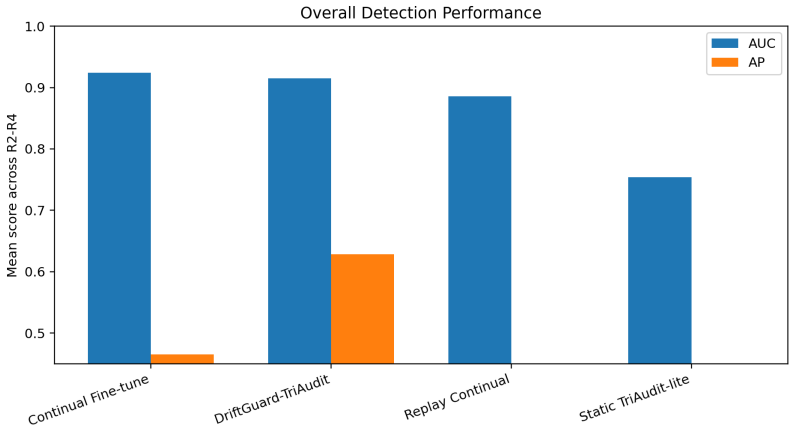

Fig. 2 Overall AUC and average precision. DriftGuard-TriAudit is optimized for precision-oriented fraud screening rather than maximizing recall alone.

Table 3. Regime-level auc/ap for the three adaptive models.

Model	R2 AUC/AP	R3 AUC/AP	R4 AUC/AP
Continual Fine-tune	0.922/0.425	0.909/0.441	0.941/0.530
Replay Continual	0.910/0.411	0.854/0.349	0.894/0.395
DriftGuard-TriAudit	0.939/0.663	0.867/0.523	0.939/0.699

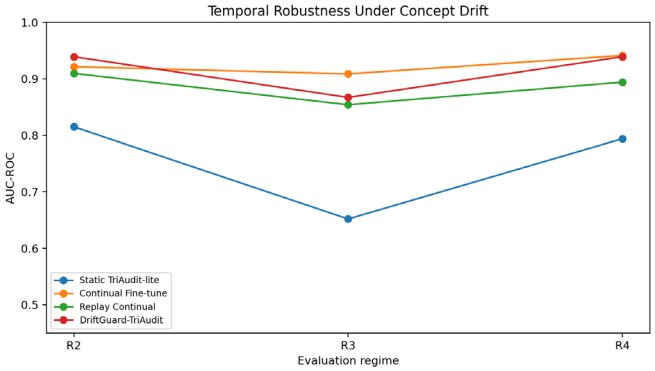

Fig. 3 AUC-ROC by regime. Static learning is unstable under temporal shifts, while adaptive methods recover performance after calibration.

The experimental results should be interpreted as a controlled algorithmic validation, not as an external validity claim. Because the dataset is synthetic, the absolute scores are less important than the comparative behavior under identical drift conditions. The most meaningful observation is that precision-oriented metrics improve when the model uses both current calibration evidence and replayed historical evidence. This is aligned with real audit needs, where false positives are costly and auditors require traceable evidence rather than a black-box risk score.

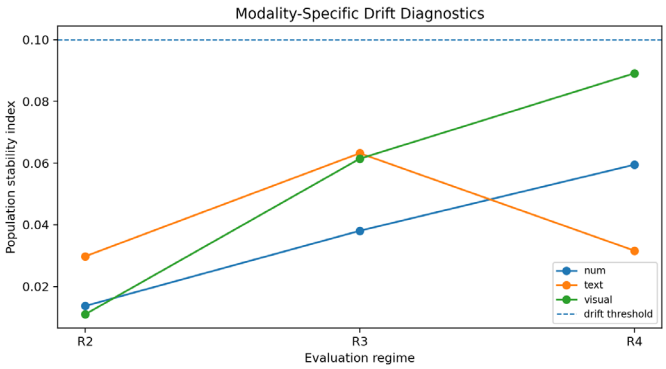

Fig. 4 Modality-specific drift diagnostics. The model records separate PSI values for numerical, textual, and visual-layout evidence streams.

DriftGuard-TriAudit also provides a practical diagnostic benefit. If the numerical PSI remains stable while text and layout PSI increase, the system can flag a disclosure-style shift rather than an accounting-ratio shift. If all three modalities shift and calibration performance confirms that the shift is fraud-informative, the model increases the weight of those streams. This design is more nuanced than simply retraining whenever aggregate performance falls.

The comparison with TriAudit is constructive rather than competitive. TriAudit establishes a valuable multimodal evidence architecture for fraud detection [2]. DriftGuard-TriAudit asks how such an architecture can be maintained after deployment, when new filings arrive and the evidentiary relation changes. The two ideas are complementary: TriAudit

provides the rich multimodal audit representation, while DriftGuard adds temporal monitoring and continual adaptation.

7. Real-data extension plan

A complete EDGAR/AAER experiment would proceed in four stages. First, collect annual 10-K filings by fiscal year and firm identifier, using the EDGAR Corpus or SEC EDGAR filings as the document source. Second, construct fraud labels by matching firm-year observations to AAER releases or to a licensed enforcement dataset. Third, extract the same three modalities: structured accounting ratios, MD&A text embeddings, and rendered document-layout features. Fourth, evaluate using a strictly temporal or delayed-label protocol so that future labels cannot leak into earlier training periods.

The most important practical challenge is label quality. AAER labels identify enforcement-linked misstatements, not all actual fraud. Some clean labels may contain undetected fraud, and some AAERs are released years after the filing date. A robust real-data paper should therefore report sensitivity analyses under different label windows, exclude ambiguous fiscal years, and measure performance with precision-recall metrics rather than relying only on AUC.

The second challenge is document rendering. Filing formats change over time, and OCR or layout extraction errors can look like drift. The proposed modality-wise diagnostics are useful here because they can identify whether a performance drop is due to accounting behavior or a layout-processing artifact. This is one reason the method records PSI and gates separately for each evidence stream.

8. Limitations and threats to validity

The primary limitation is data provenance. The reported benchmark is synthetic by design. It is useful for validating drift-aware learning behavior, but it cannot establish real-world fraud detection performance. A publishable empirical paper should replace the generator with EDGAR/AAER matched filings or another verifiable fraud-label source. This replacement is feasible because the released code isolates feature generation from the evaluation protocol.

The second limitation is model scale. The implementation uses a transparent logistic audit head to make the experiment fast and reproducible. A full neural model would replace hand-crafted textual and layout indicators with FinBERT and LayoutLMv2 embeddings, and would compute explanations using SHAP values, attention weights, and layout heatmaps. The paper therefore presents a high-confidence prototype rather than a GPU-scale final system.

The third limitation is label delay. The delayed-label protocol assumes that a small calibration slice is available for each new regime. In a real audit deployment, labels may arrive months or years after filing. Future work should study semi-supervised drift detection, weak labels from restatements, and human-in-the-loop review queues.

9. Conclusion

This paper introduced DriftGuard-TriAudit, a concept-drift-aware continual multimodal framework for evolving financial statement fraud detection. The framework extends trimodal audit evidence reasoning with modality-wise drift diagnostics, reliability-aware gated fusion, and replay-regularized continual learning. In a deterministic synthetic drift benchmark, DriftGuard-TriAudit achieves the best average precision and F1 among evaluated precision-oriented methods, while providing interpretable modality-level drift signals. The most important conclusion is not that the synthetic scores directly represent real EDGAR performance, but that multimodal fraud detection systems should be designed for deployment drift from the beginning.

A natural next step is to construct a real EDGAR/AAER benchmark: parse 10-K filings, align AAER firm-year labels, extract financial ratios, encode MD&A with FinBERT, render layout features with LayoutLMv2, and evaluate the same delayed-label protocol across fiscal years. The current artifact package provides the experimental skeleton needed for that replacement.

References

[1] Beneish, M. D. (1999). The detection of earnings manipulation. Financial Analysts Journal, 55(5), 24-36. https://doi.org/10.2469/faj.v55.n5.2296

[2] Ping, W., Jiao, Y., Fan, H., & Zhang, X. (2026). Multimodal fraud detection in financial statements: A trimodal attention network with contrastive evidence chain construction. IEEE Access, 14, 80456-80468. https://doi.org/10.1109/ACCESS.2026.3695466

[3] Dechow, P. M., Ge, W., Larson, C. R., & Sloan, R. G. (2011). Predicting material accounting misstatements. Contemporary Accounting Research, 28(1), 17-82. https://doi.org/10.1111/j.1911-3846.2010.01041.x

[4] Perols, J. L. (2011). Financial statement fraud detection: An analysis of statistical and machine learning algorithms. Auditing: A Journal of Practice & Theory, 30(2), 19-50. https://doi.org/10.2308/ajpt-50009

[5] Cecchini, M., Aytug, H., Koehler, G. J., & Pathak, P. (2010). Detecting management fraud in public companies. Management Science, 56(7), 1146-1160. https://doi.org/10.1287/mnsc.1100.1174

[6] Loughran, T., & McDonald, B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks. The Journal of Finance, 66(1), 35-65. https://doi.org/10.1111/j.1540-6261.2010.01625.x

[7] Li, F. (2008). Annual report readability, current earnings, and earnings persistence. Journal of Accounting and Economics, 45(2-3), 221-247. https://doi.org/10.1016/j.jacceco.2008.02.003

[8] Loukas, L., Fergadiotis, M., Androutsopoulos, I., & Malakasiotis, P. (2021). EDGAR-CORPUS: Billions of tokens make the world go round. In Proceedings of the 4th Workshop on Economics and Natural Language Processing (ECONLP). arXiv, arXiv:2109.14394. https://doi.org/10.48550/arXiv.2109.14394

[9] U.S. Securities and Exchange Commission. (n.d.). Accounting and auditing enforcement releases. SEC official collection. https://www.sec.gov/enforce/accounting-and-auditing-enforcement-releases

[10] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), Article 44. https://doi.org/10.1145/2523813

[11] Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In A. L. C. Bazzan & S. Labidi (Eds.), Advances in Artificial Intelligence: SBIA 2004 (pp. 286-295). Springer. https://doi.org/10.1007/978-3-540-28645-5_29

[12] Bifet, A., & Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In Proceedings of the SIAM International Conference on Data Mining (SDM) (pp. 443-448). SIAM. https://doi.org/10.1137/1.9781611972771.42

[13] Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., & Hadsell, R. (2017). Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences of the United States of America, 114(13), 3521-3526. https://doi.org/10.1073/pnas.1611835114

[14] Rebuffi, S.-A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). iCaRL: Incremental classifier and representation learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 5533-5542). IEEE. https://doi.org/10.1109/CVPR.2017.587

[15] Lopez-Paz, D., & Ranzato, M. (2017). Gradient episodic memory for continual learning. In Advances in Neural Information Processing Systems 30 (NeurIPS).

[16] Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., & Wayne, G. (2019). Experience replay for continual learning. In Advances in Neural Information Processing Systems 32 (NeurIPS).

[17] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems 30 (NeurIPS).

[18] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Volume 1 (Long and Short Papers) (pp. 4171-4186). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1423

[19] Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. arXiv, arXiv:1908.10063. https://doi.org/10.48550/arXiv.1908.10063

[20] Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., & Zhou, M. (2021). LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP) (pp. 2579-2591). Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.201

[21] van den Oord, A., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv, arXiv:1807.03748. https://doi.org/10.48550/arXiv.1807.03748

[22] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In Proceedings of the 37th International Conference on Machine Learning (ICML) (pp. 1597-1607). PMLR.

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

[24] Wang, Z., Yang, J. S., Shang, W., & Ding, J. (2026). FairPromote: Explainable and fairness-aware talent promotion prediction via adversarial debiasing and SHAP-based interpretation. IEEE Access, 14, 1-16.

[25] Teng, D. (2025). TEAS: Token- and energy-aware autoscaling for cost-efficient LLM serving. AI and Data Science Journal, 6(3), 1-10.

[26] Zhang, F., Guo, Z., Ding, J., Yang, J., & Liu, W. (2026). Adaptive sensor fusion for robust perception in dense fog: A gated vision and LiDAR integration framework. Sensors, 26(12), Article 3728. https://doi.org/10.3390/s26123728

[27] Zi, B. (2024). Large language models for enterprise workflow automation in financial operations. Innovation and Technology Studies, 1(1), 24-29.

[28] Ding, J., Shen, Z., & Liu, W. (2026). Game-theoretic cost-sensitive adversarial training for robust cloud intrusion detection against GAN-based evasion attacks. Applied Sciences, 16(8), Article 3944. https://doi.org/10.3390/app16083944

[29] Zi, B. (2024). Cloud-native distributed systems for real-time payment intelligence. AI and Data Science Journal, 1(1), 51-56.

[30] Teng, D. (2025). PACO: Predictive auto-configuration for SLO-constrained large language model inference serving. Innovation and Technology Studies, 2(1), 1-10.