How Artificial Intelligence Enriches the Traditional Data Science Toolbox in Finance

Cheng Cui*

Department of Statistics, College of Literature, Science, and the Arts, University of Michigan-Ann Arbor, 48109 Ann Arbor, MI, USA

Abstract. With the continuous growth of the scale, heterogeneity and dynamics of financial data, traditional statistical and machine learning methods have gradually shown deficiencies in feature-engineering dependence, nonlinear modeling and stability in rapidly changing environments. Artificial intelligence (AI) has provided new tools and paradigms for financial data science, significantly enhancing performance in core tasks such as credit risk modeling, fraud detection, portfolio optimization, and market forecasting. Specifically, the three main lines of representation learning, relational learning, and time-series and real-time learning have enriched the modeling methods of data science, while decision engineering and governance & compliance have become the two major guarantees. However, in the process of implementation, AI still faces challenges such as data availability and out-of-distribution generalization/transferability, compliance-grade explainability and computational cost, probability calibration and decision engineering, dual sensitivity to data/structural drift and objective/hyperparameter design, large-scale deployment and latency, and the sustainability of returns and execution feasibility. This paper systematically reviews the expansion of the traditional data science toolkit by AI in the financial field and looks forward to the potential development paths in the future in areas such as financial foundation models, the integration of causal inference and constrained decision-making, and temporal/dynamic graphs and multi-agent market simulation.

*Corresponding author: chengcui@umich.edu

1 Introduction

In the modern financial industry, data-driven decision-making has become the core of many businesses such as risk management, credit assessment and market forecasting. However, as financial data continues to grow in terms of scale, heterogeneity and dynamics, some traditional statistical or machine learning methods are gradually revealing their limitations: feature engineering is highly dependent on the experience of domain experts, nonlinear relationships are difficult to model, and models are also prone to failure in a rapidly changing market environment.

At present, it can be observed that there have been many examples where Data Science has further expanded its influence in the field of Finance with the help of AI. For instance, in terms of credit scoring and default prediction. Hjelkrem and de Lange explored a deep learning credit scoring model based on open banking transaction texts. They found that deep learning models achieved better performance in predicting defaults [1]. Liu et al. proposed Local–Global Graph Neural Network (LG-GNN) for credit scoring by combining local–global graph aggregation with contrastive learning. They found it outperformed traditional baselines and strong competitors across multiple datasets [2]. Furthermore, there has also been a significant improvement in Fraud Detection. Fanai and Abbasimehr used a two-stage framework that first learns representations with a deep autoencoder (AE) and then trains deep classifiers for credit card fraud detection. They found AE-based models consistently beat their baselines and also outperformed principal component analysis (PCA)-based counterparts [3]. Harish et al. applied a relational graph convolutional network to a heterogeneous transaction graph for fraud detection. They found the graph neural network (GNN) leveraged neighborhood information to outperform classical ML and standard deep learning baselines [4]. Meanwhile, this improvement is also reflected in the aspect of portfolio optimization. Lim, Cao, and Quek applied reinforcement learning to dynamic portfolio rebalancing, with and without an long short-term memory (LSTM) price-prediction module. They found the reinforcement learning (RL) approach with gradual rebalancing and prediction delivered the best portfolio performance versus alternatives [5]. Jiang, Olmo, and Atwi employed Twin Delayed Deep Deterministic Policy Gradient (TD3)-style deep reinforcement learning for high-dimensional portfolio allocation under risk and transaction-cost constraints. They found it outperformed conventional allocation rules and common deep reinforcement learning (DRL) baselines out of sample [6]. It is also reflected in the aspects of Market Prediction/Sentiment & Volatility.

Kim and Won integrated LSTM with generalized autoregressive conditional heteroskedasticity (GARCH)-type models to forecast volatility. They found the hybrid reduced forecast errors and significantly outperformed existing econometric and deep learning baselines [7]. Fischer and Krauss used LSTM networks for daily directional prediction/statistical arbitrage on Standard & Poor's (S&P) 500 constituents. They found LSTM achieved higher out-of-sample returns and predictive accuracy than random forest, a standard deep net, and logistic regression [8].

This article aims to review how AI enriches the toolbox of traditional data science in the financial field, optimizing the accuracy and efficiency of existing methods while also promoting the continuous evolution of financial data science in addressing complex data and modeling challenges. This study will focus on four core tasks - credit risk modeling, fraud detection, portfolio optimization and market forecasting, explore the complementary relationship between AI technology and existing data science methods, and analyze the challenges and future development directions brought about by this. Through this review, this study hopes to present a more comprehensive picture: Financial data science is undergoing an AI-driven evolution, and this evolution is not about substitution but rather integration and collaboration.

2 AI Applications across Financial Domains

Overall, the AI benefits of DS in the financial field can be summarized as "three main lines +two safeguards". Three main lines: Representation learning (self-supervised/contrastive, extracting nonlinear features from raw multimodal data, avoiding simply summarizing time series into a few static indicators); Relational learning (extending the independent-sample assumption to graph-based dependence, characterizing networks and contagion effects using GNN/subgraph retrieval); Time-series and real-time (sequence/stream modeling, online detection and drift adaptation). Two safeguards: decision engineering (utility/constraint awareness, robust calibration/thresholding, cost/capacity, from cascading inference to linkage to execution) and governance & compliance (rigorous backtesting/data-leakage prevention, data/feature lineage, explainability/adverse-action explanations, robustness/fairness/drift monitoring and compliance boundaries).

2.1 Credit risk modeling

Firstly, representation learning. In their research, Hjelkrem and de Lange [1] delved into the feasibility and value of directly utilizing the textual descriptions of raw transactions provided by open banking APIs in the field of credit risk modeling. This study directly compares two AI approaches: one is a lightweight deep text classifier trained from scratch (vectorization, word embedding, CNN, global max pooling, and dropout), and the other is a multilingual BERT based on transfer learning and fine-tuning. The study adopts real-world retail credit data from a medium-sized bank in Norway. Data processing concatenates all transaction descriptions in the 90 days prior to the scoring date into a single long text to predict default within the following 12 months; class imbalance is handled by undersampling. The key finding is that across training, validation, and test sets, the from-scratch CNN achieved higher AUC and lower Brier than the fine-tuned multilingual BERT; on the out-of-time test set, it reached an AUC of 0.909 (Brier 0.015) versus 0.825 (Brier 0.023) for BERT. Meanwhile, the study uses SHAP for global and

individual interpretation on 1,166 high-risk samples (predicted default rate ⩾ 25%), identifying high-contribution terms such as restanse (debt), forsinkelsesrente (late payment fee), debetrenter (debit interest), udekning (insufficient funds), mislighold (default), and NAV; waterfall plots support explanations of individual loan-rejection cases—achieving a practical balance between "performance improvement" and "compliance-grade explainability."

Secondly, in terms of relationship modeling. Liu et al. [2] proposed LG-GNN, which, for default detection/credit scoring, jointly models the local neighborhood and global structure and introduces supervised contrastive learning to enhance class separability: on seven real data subsets (LendingClub×3, HomeCredit×2, PPD×2), samples are treated as nodes and a cosine-similarity graph is constructed with an edge-threshold to control density. Subsequently, for each node, Top-K local neighbors are selected via attention scoring, and Top-K global neighbors are selected directly based on attention coefficients (avoiding reliance on high-order adjacency). The model first refines the two types of neighborhood signals through Local Information Filtering (LIF) and Global Information Filtering (GIF), then adaptively fuses them with learnable weights to obtain the node representation. Combined with a supervised contrastive loss (bringing same-class nodes closer and pushing different-class nodes apart) and an MLP classification head, it outputs the default probability. Compared with traditional/existing methods, its innovation lies in: local + global dual-perspective filtering to suppress noise, adaptive fusion to avoid information mis-propagation from naïve concatenation, and task-aware supervised contrastive learning with end-to-end graph representation to reduce heavy manual feature engineering. Experiments show that across seven datasets, average improvements over traditional baselines are +18.05% / +25.76% / +9.62% / +133.46% (AUC / F-score / ACC / KS), and over the second-best method are +1.73% / +1.18% / +1.44% / +26.51%. On the three LendingClub subsets, improvements over all baselines are AUC +2.52%, F1 +12.36%, Acc +1.93%; relative to the second-best method: AUC +1.34%, F1 +0.16%, Acc +0.08%.

2.2 Fraud detection

In the “Representation Learning” main line, Fanai & Abbasimehr [3] proposed a two-stage framework for credit card fraud detection. Stage one uses a deep autoencoder (AE) to learn low-dimensional representations from the original features (for the European Cardholders dataset, the feature space was reduced from 29 to 22 dimensions). Stage two feeds these AE representations into multiple supervised classifiers (DNN / RNN / CNN_RNN, with SVM as a baseline comparator), and systematically compares performance against both the original inputs and PCA-based representations. The study covers two benchmark datasets—European Cardholders and German Credit—both imbalanced (severely so for European Cardholders). Compared with traditional approaches, the key innovation is to use deep representations as a model-agnostic front end,

showing that AE-based features deliver consistent gains across classifiers and outperform PCA at the same dimensionality. Representative holdout results: on the European dataset, from AE to CNN_RNN achieved the best overall performance, surpassing Fiore et al. (2019)'s GAN baseline in F1 (0.8372 vs. 0.8205); on German Credit, from AE to DNN likewise showed improvements under cross-validation. Overall conclusion: using an AE for representation learning/dimensionality reduction reliably improves supervised fraud detection in highly imbalanced settings .

On "Relational Learning" track, Sunisha Harish et al. [4] construct a heterogeneous graph for credit-card fraud detection and use an R-GCN to perform transaction-node classification, benchmarking against a suite of classical ML and DL baselines. Methodologically, they take the transaction as the core node (retaining all transaction features), model card ID and merchant ID as featureless nodes with learnable embeddings, and connect nodes via two relation types: card-to-transaction and transaction-to-merchant. Two highly imbalanced benchmark datasets (IBM TabFormer, fraud is approximately 0.1%; Sparkov) are used; after a unified preprocessing pipeline, models are trained without class-resampling. Compared with traditional approaches, the innovations are: explicitly encoding the ternary card–transaction–merchant relations with a heterogeneous graph while keeping transactions as nodes to avoid metadata loss; and using R-GCN's relation-type-specific weights for message passing to aggregate more relevant local-neighborhood information. Overall, this graph formulation is more general and transferable than manual-rule or homogeneous-graph setups. Results: on IBM TabFormer, R-GCN attains F1-score 0.78 and Recall 0.66, a small improvement over the best classical baseline (CatBoost/XGBoost, F1 0.76), with AUCPR 0.78 slightly below 0.80 for CatBoost/XGBoost; on Sparkov, R-GCN achieves F1 0.61 and Recall 0.46, outperforming the best DL baseline (LSTM, F1 0.58) and the best classical baseline (XGBoost, F1 0.39), with AUCPR 0.66 exceeding LSTM (0.64) and XGBoost (0.48). Overall, on datasets with realistic business structure and extreme class imbalance, graph learning (R-GCN) outperforms classical ML and common deep sequence models, validating the value of explicitly modeling local-neighborhood relations in transaction networks.

2.3 Portfolio optimization

This is a case that falls under the "Time-series and Real-time" track. Lim et al.[5] combine trend-type time-series signals (EMA/MACD) with a reinforcement-learning decision loop and adopt progressive (rather than one-shot) rebalancing, alongside SAA+TAA to maintain diversification under a 0.125% transaction-cost constraint. An LSTM is used to forecast short-term prices to mitigate indicator lag, making weight adjustments more forward-looking. Methodologically, the state is composed of EMA/MACD-based differences for two assets and the number of days since the last reversal (Dt); the action reallocates weights among high/medium/low-risk assets; and the reward combines the 10-day-ahead

change in NAV relative to the baseline (with time scaling) plus a normalized current NAV term. Across three asset baskets (Global Index group, U.S. large-cap group, and Nasdaq small/mid-cap group), compared with a baseline of one-shot full rebalancing without prediction, the "progressive rebalancing + LSTM" setup yields roughly 27.9%–93.4% higher end-of-period NAV and materially improves maximum drawdown. For example, in the Global Index group it achieves a 55.2% terminal return with a 0.7% maximum drawdown; in the U.S. large-cap group, the combined portfolio return exceeds any single constituent—corroborating the effectiveness of the "trend signal + short-term prediction + progressive online rebalancing" approach.

This case also falls under the "Time-series and Real-time" track and emphasizes decision engineering. Jiang et al. [6] directly embed investor risk aversion and transaction costs into an extended mean–variance-type reward function (portfolio-value term - risk penalty - transaction cost) and perform dynamic rebalancing with TD3 in a continuous action space. The policy network uses CNN/LSTM, and overestimation of Q-values is mitigated—while stability and generalization across high-dimensional asset cross-sections are improved—via twin critics, target policy smoothing, and delayed policy updates. Compared with Max-Sharpe, minimum-variance (MV), and DRL baselines (PPO, DDPG, standard TD3), it achieves a more balanced performance across the three dimensions of return, risk, and cost. In terms of workflow, portfolio trading is formulated as an MDP: the state includes prices and positions; actions are buy/sell/hold (extendable to continuous weight adjustments); and the reward is the portfolio-value term minus risk and transaction-cost penalties. Using Yahoo Finance data for the DJIA 30 and S&P 100, the study conducts segmented training/validation/out-of-time testing and evaluates with annualized/cumulative return, annualized volatility, Sharpe, maximum drawdown (MDD), and Calmar ratio against the aforementioned baselines. The result is that, on DJIA, RTC-CNN-TD3 delivers higher annualized return and Sharpe than Max-Sharpe and MV; on S&P 100, RTC-LSTM-TD3 and RTC-CNN-TD3 likewise outperform both traditional and DRL baselines on return and Sharpe. Sensitivity analysis shows that raising transaction costs significantly compresses returns (e.g., increasing the cost rate from 0.01% to 0.5% materially reduces cumulative return), while increasing risk aversion lowers volatility and drawdown at the expense of return—confirming the method's tunability and robustness under the return–risk–cost trade-off.

2.4 Market forecasting

These cases all fall under the "Time-series and Real-time" track. Kim and Won [7] propose a hybrid framework that combines LSTM with multiple GARCH-type models for stock-index volatility forecasting; unlike approaches that blend a neural network with a single econometric model, the authors systematically construct a family of "LSTM + 1–3 GARCH-type models" (including GARCH, EGARCH, and EWMA) to

jointly capture sequence-pattern learning and conditional heteroskedasticity. Using KOSPI 200 data, they train single-model baselines (GARCH, EGARCH, EWMA, a deep feedforward network, and LSTM), a DFN+GARCH hybrid baseline, and several LSTM-based hybrids (e.g., E-LSTM, GE-LSTM, GEW-LSTM), and compare performance using MAE, MSE, HMAE, and HMSE. Robustness is assessed across 7/15/22-day input windows and 1/14/21-day forecast horizons, with out-of-sample differences evaluated via forecast-accuracy/equality tests such as Diebold–Mariano (DM) and the Wilcoxon signed-rank (WS) test. Results show that the best GEW-LSTM (LSTM+GARCH+EGARCH+EWMA) attains MAE = 0.0107 on KOSPI 200, a 37.2% reduction versus the then-strong E-DFN baseline (EGARCH+DFN, MAE = 0.017); MSE/HMAE/HMSE decrease by 57.3%/24.7%/48%, and GEW-LSTM maintains the lowest errors across windows and horizons, outperforming both single models and the DFN hybrid baseline.

Fischer and Krauss [8] propose using an LSTM for cross-sectional next-day direction prediction and linking the forecasts to trade execution. The innovation over memory-independent classifiers (random forest, deep feedforward neural network, logistic regression) is to perform memory-based sequence learning on a 240-day rolling return window to capture long-term dependencies, and to provide interpretable attribution of profit sources (high volatility plus short-term reversal characteristics). They further construct a reproducible, rules-based strategy as a comparator derived from these patterns; in addition, they run robustness checks (VWAP pricing, weekly frequency, and a one-day signal-lag rule) and use Diebold–Mariano (DM) and Pesaran–Timmermann (PT) tests to assess predictive superiority. In terms of workflow, each asset's input feature is a standardized series of daily returns (length 240), with a binary target of whether it outperforms the cross-sectional median on the next day. Model outputs rank the day's ~500 S&P 500 constituents by predicted probability to form a long top-k / short bottom-k long–short portfolio, progressing through 1992–2015 in rolling windows of 750 trading days for training / 250 for trading, and incorporating a one-way 5 bps transaction cost to evaluate returns, volatility, Sharpe, and drawdowns. Results show the LSTM achieves an average daily return of 0.46% (pre-cost) / 0.26% (post-cost), exceeding random forest (0.43%/0.23%), deep feedforward network (0.32%/0.12%), and logistic regression (0.26%/0.06%), with higher predictive accuracy and a post-cost annualized Sharpe of ~2.34. In attribution, selected stocks typically exhibit short-term extreme moves followed by reversals and higher volatility; a simple rules strategy of "buy short-term extreme losers / sell extreme winners" delivers only ~0.23% per day pre-cost—about half the LSTM's profit—indicating the LSTM extracts richer nonlinear structure from the sequences. These conclusions remain statistically and economically significant under VWAP pricing and weekly rebalancing.

3 Discussion

3.1 Challenges

Although AI technology has significantly improved the performance of various financial tasks, it still faces the following types of challenges.

The first is data availability and out-of-distribution generalization/transferability. Take the research of Hjelkrem & de Lange as an example. Since it is only based on one Norwegian bank and includes an out-of-time test, this makes it still uncertain in terms of cross-customer group, cross-institutional or cross-border migration. Meanwhile, the differences in language and reporting definitions (register/schema) of open banking, combined with reject inference (unlabeled rejected samples), can easily lead to a deterioration in the performance of the model after its launch.

The second challenge is the tension between compliance-grade explainability and computational cost. Accurately calculating SHAP in text or graph scenarios consumes a lot of computing resources. Due to the limitation of computing power, these studies usually only explain a small number of high-risk samples and also use approximate explainers (e.g., SHAP's partition explainer / Owen-value approximation), which brings certain uncertainties in interpretation. Moreover, the interpretation tools for graph models (such as LG-GNN) at the node or subgraph-level are not yet mature, making it difficult to simultaneously achieve explanations that are understandable to customers without disclosing the privacy information of others.

The third challenge is probability calibration and decision engineering. The improvement of ranking/discrimination capabilities does not equal business gains. Therefore, it is necessary to maintain stable calibration for different groups of people and different cycles, and to maintain threshold and cost-sensitive strategies in the long term.

The fourth challenge is the dual sensitivity to the environment and objective design. The model is sensitive to both external data/structure drift and internal target and hyperparameter settings. Graph methods (such as LG-GNN) are highly sensitive to the graph construction choices (e.g., edge-building criteria), similarity threshold, Top-K, and local/global filtering; once the customer structure or the macro environment changes, the graph structure will be significantly modified, resulting in the neighborhood/distribution that the model most relies on no longer matching. When the text side switches to a new statement system/register, the performance of the general pre-trained model may also deteriorate. At the same time, it is a reasonable direction to directly incorporate transaction costs and risk aversion into the objective function and strategy, but this will bring risks such as hyperparameter sensitivity and the fact that the offline optimum may not be reproducible online. For instance, Jiang - Olmo - Atwi shows that increasing costs will significantly compress returns; enhancing risk aversion can reduce volatility and drawdowns, but it will sacrifice returns.

The fifth challenge is that it is difficult to achieve both large-scale deployment and latency. Modeling directly with the original text and then explaining it afterwards, or representing it with complex graphs and

updating it online, are both highly sensitive to computing power and latency. When combined with data minimization, cross-border data transfer and fairness governance, institutions can only constantly balance among timeliness, cost, experience and compliance. Graph models also have to confront operational and maintenance thresholds such as online incremental graph construction, cold start, de-noising, and poisoning-defense.

The sixth challenge is the sustainability of returns and the gap in execution. Once the signal is widely used, the excess can be easily smoothed out by market arbitrage and is also affected by the trading microstructure and slippage. For instance, Fischer & Krauss saw a significant decline in returns after using VWAP (closer to actual executions) and after switching to weekly frequency and a one-day signal lag. Although Lim et al.'s "progressive rebalancing + short-term prediction" can improve to some extent, it is still very sensitive to transaction costs and latency.

3.2 Future prospects

Although there are still many problems, the future prospects are still worth looking forward to.

For data, the availability and the out-of-distribution generalization/transferability, can learn by federated learning/SMPC (secure multiparty computation) privacy-friendly cooperation development samples cover [9], At the same time, promote the standardization of reporting definitions and ontology (for example, align transaction texts with merchant/account entities), and combine domain adaptation/continued pretraining (DAPT/TAPT) and OOD detection + PU (positive – unlabeled)/semi-supervised learning to alleviate reject inference and cross-domain drift.

For compliance-grade explainability and computational cost, the explainability is moved forward to the design constraint, and the "deep representation + explainable decision head" (such as LR/GAM/monotonic GBDT; The tree model is preferentially implemented using the exact/near-linear-time complexity of TreeSHAP) [10], and most of the explanations are computed in offline batches and cached as standardized reason codes and evidence. Trigger high-cost interpretations only for boundary/anomaly/check samples online, prioritize the use of verifiable approximations (sampling/Partition SHAP), set fidelity thresholds and provide fallback to precise methods; graph setting matching GNNExplainer/PGExplainer/GraphSHAP do training attribution, outputs do desensitization and privacy audit evidence.

For probability calibration and decision engineering, cluster calibration (isotonic/Platt/temperature scaling) is performed by customer group/period, and the uncertainty interval is given by conformal prediction. Transform the steady state of ranking advantage into an executable strategy by minimizing the cost-sensitive threshold and expected loss, and combine it with the champion - challenger and online recalibration closed loop.

For dual sensitivity to the environment and objective design, on the graph side, time decay/sliding windows or learnable graph construction with neighborhood stability regularization are adopted, and on the text side, intra-domain continued pretraining and term alignment are performed. At the same time, transaction costs and risk aversion are included in the target, but hyperparameter sensitivity analysis, robust regularization and Bayesian/AutoML hyperparameter tuning are provided to reduce the risk that "the best offline is difficult to go online".

For the difficulty of achieving both large-scale deployment and latency, control the latency through distillation/quantization/pruning and cascading inference, build a real-time feature library and graph cache to support incremental graph construction and neighbor indexing And embed data minimization, cross-border and fairness monitoring into MLOps (CI/CD, drift monitoring, automatic rollback and explanation governance). For sustainability of returns and the gap in execution, realistic constraints such as VWAP/slippage/cost/capacity are pre-imposed in the research stage and signal lag tests are conducted. At the trading level, progressive rebalancing, cost/shock prediction and gradual ramp-up are adopted. Combining intelligent routing with congestion monitoring, it ensures that excess is reproducible and sustainable in actual execution.

Overall, if taking it one step further, (1) the financial foundation model: by using domain corpora + structured statements/graphs + time series for joint self-supervision, and superimposed differential privacy/TEE for training and RAG-style controlled retrieval, it not only improves representation but also controls leakage. (2) Integration of causality and constrained decision-making: Link the discovery of explained causality/counterfactual inference with differentiable constraint layers (such as monotonic/budget/capacity/ESG) or CVaR/Safe-RL to directly internalize the multi-objective trade-off of "benefit - risk - cost - constraint" within the training objective. (3) Dynamic/temporal graph GNNs and multi-agent market simulation, using temporal graph GNN to learn the evolution of relationships over time, and conducting strategy robustness and congestion/shock tests in the multi-agent microstructure simulator. These technologies, which are not yet fully mature or widely adopted, all hold great promise to substantially alleviate the aforementioned bottlenecks within the boundaries of engineering feasibility and compliance. In any case, the integration of AI and the DS toolbox is an inevitable trend in the future.

4 Conclusion

In conclusion, AI is driving financial data science into a new stage of "enhancement rather than replacement": in areas such as credit risk, fraud detection, portfolio and market forecasting, AI methods have significantly enhanced the capabilities of traditional toolboxes in nonlinear feature extraction, complex relationship modeling and time-series forecasting. However, it still faces challenges in terms of data, interpretability, stability and deployment. Looking ahead, with the development of technologies

such as financial native large models, causal inference and constrained optimization, dynamic graph learning and multi-agent simulation, AI is expected to achieve more stable, sustainable and scalable financial applications within the compliance boundary.

References

1. L.O. Hjelkrem, P.E. de Lange, Explaining deep learning models for credit scoring with SHAP: A case study using open banking data. J. Risk Financ. Manag. 16, 221 (2023).
2. Y. Liu, X. Wang, T. Meng, W. Ai, K. Li, LG-GNN: Local and global information-aware graph neural network for default detection. Comput. Oper. Res. 169, 106738 (2024).
3. H. Fanai, H. Abbasimehr, A novel combined approach based on deep autoencoder and deep classifiers for credit card fraud detection. Expert Syst. Appl. 217, 119562 (2023).
4. S. Harish, C. Lakhanpal, A.H. Jafari, Leveraging graph-based learning for credit card fraud detection: A comparative study of classical, deep learning and graph-based approaches. Neural Comput. Appl. 36, 21873–21883 (2024).
5. Q.Y.E. Lim, Q. Cao, C. Quek, Dynamic portfolio rebalancing through reinforcement learning. Neural Comput. Appl. 34, 7125–7139 (2022).
6. Y. Jiang, J. Olmo, M. Atwi, Deep reinforcement learning for portfolio selection. Global Finance J. 62, 101016 (2024).
7. H.Y. Kim, C.H. Won, Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. Expert Syst. Appl. 103, 25–37 (2018).
8. T. Fischer, C. Krauss, Deep learning with long short-term memory networks for financial market predictions. Eur. J. Oper. Res. 270, 654–669 (2018). https://doi.org/10.1016/j.ejor.2017.11.054
9. A. Ahmed, M.A. Javed, J.N. Qureshi, H. Khan, H.F. Yousaf, An insightful machine learning based privacy-preserving technique for federated learning. Asian Bull. Big Data Manag. 4, 332–343 (2024)
10. J. Yang, Fast treeshap: Accelerating SHAP value computation for trees. arXiv:2109.09847 (2021)