Proceedings of CONF-BPS 2026 Symposium: GenAI, Labour Markets, and the Economics of Human and Financial Capital
DOI: 10.54254/2754-1169/2026.LD31802

From Prediction to Decision: A Survey of Machine Learning Applications in Quantitative Finance

Kintat Ng

Faculty of Mathematics & Physical Science, University College London, London, UK
ZCAHKNG@ucl.ac.uk

Abstract. With the explosive growth of computer science and big data, quantitative finance is undergoing a huge shift from traditional econometrics to data-driven Artificial Intelligence (AI). This study aims to review the applications of Machine Learning (ML), Deep Learning (DL), and Reinforcement Learning (RL) in vital areas such as asset pricing, algorithmic trading and risk management. Research shows that Deep Neural Networks (DNNs) capture non-linear market patterns, Natural Language Processing (NLP) analyses unstructured data while providing superior sentiment signals compared to generic alternatives. Deep Reinforcement Learning (DRL) can effectively automate execution in dynamic market environments, which optimizes trading decisions. Similarly, Temporal Fusion Transformers (TFT) have emerged as a dominant architecture for multi-horizon time series forecasting, offering superior interpretability and accuracy over standard recurrent networks. These ML models greatly outperform traditional methods like CAPM and ARIMA in prediction accuracy and handling complex data. However, there are still challenges about the lack of interpretability ("black box") and overfitting. In the future, quantitative finance lies in combining "Explainable AI" (XAI).

Keywords: Quantitative Finance, Asset Pricing, Deep Learning, Machine Learning

1. Introduction

Financial markets are known for little clear signal compared to random noise. Also, markets keep changing over time and show complex, non-straightforward patterns. The older financial models that depend mostly on simple straight-line relationships—for example, the well-known Fama-French factor models—are finding it harder and harder to deal with today's fast-moving and complicated markets.

Fortunately, machine learning has appeared in recent years as a very useful set of tools. These methods are good at handling huge amounts of information and finding hidden non-linear connections that traditional approaches often miss.

This paper looks carefully at the ways machine learning is changing the full process of quantitative investing. It covers everything from searching for new factors that explain returns to building systems that automatically carry out trades. In academic work, the focus has shifted over time. Researchers have moved away from trying to predict just tomorrow's prices towards developing complete systems that make trading decisions from start to finish. Some interesting

DOI: 10.54254/2754-1169/2026.LD31802

recent examples are using Transformer models to forecast prices across different time periods, and applying a version of BERT called FinBERT to understand what central banks are really saying in their official statements.

This research uses a Systematic Literature Review method to examine published studies and clearly answer three main questions: (1) Are deep learning approaches better than classic factor models when it comes to explaining and predicting asset prices? (2) How well does Reinforcement Learning work when people try to use it for real trading in live markets? (3) What practical solutions exist to make these complex machine learning models easier to understand and explain, so they can satisfy the usual regulatory and industry requirements?

2. The paradigm shift: from econometrics to machine learning

2.1. Limitations of linear models

For decades, the main way people did empirical asset pricing was with linear factor models. These come from the Arbitrage Pricing Theory (APT). The idea is simple: an asset's expected excess return is just a straight-line combination of how sensitive (beta) it is to various systematic risk factors. CAPM said the only factor that mattered was the market portfolio. Then Fama and French added size (SMB) and value (HML), and later they brought in profitability (RMW) and investment (CMA) too.

But these days, the strict linear setup has turned into a real problem in current markets. The biggest issue is that everything is forced into a linear shape. The models assume a predictor like the book-to-market ratio has the same effect on expected returns, no matter what else is going on. That almost never holds. Take "value" signals: they can work great in calm low-volatility times but become useless or even harmful during high-volatility periods. Normal linear regression can't handle those kinds of changing relationships unless you add extra interaction terms by hand – and once you have more than a handful of variables, doing that manually becomes impossible.

2.2. ML advantage and the bias-variance tradeoff

Going to machine learning basically changes how you deal with the bias-variance tradeoff. In any statistical learning setup, the total error when predicting new data comes from two things: bias and variance.

Bias is the error built in from starting assumptions that are wrong—like saying all relationships must be linear when they aren't. Variance is the error from the model latching onto every little wiggle and random bit in the exact training data you happened to use.

• Linear Models (High Bias, Low Variance): Traditional econometric models carry high bias because they lock in strong assumptions like linearity that frequently don't fit real financial markets. On the good side, they have low variance—keep things so basic that swapping one data sample for another doesn't change the answers much.

• Machine Learning (Low Bias, High Variance): ML models—particularly deep neural networks (DNNs)—can act as universal function approximators. They learn very complicated non-linear patterns straight from the data and get bias almost to zero. Back in the day, the catch was high variance: they overfit everything and memorized the training set. But now with way bigger datasets plus regularization methods (dropout, L1/L2 penalties and so on), it can take the variance and keep the low-bias benefit.

DOI: 10.54254/2754-1169/2026.LD31802

Heaton, Polson, and Witte show that deep learning spots interactions in financial data that standard economic theory never even thought about. When you get the bias-variance balance right, machine learning uncovers "deep" factors that linear models are built to miss completely [1].

3. Asset pricing and factor mining

3.1. Deep learning and stochastic discount factor

The core idea in asset pricing is the Stochastic Discount Factor (SDF), sometimes called the pricing kernel. The basic theorem says any asset's price equals the expected value of its future payoffs, discounted by this SDF. Older models usually tried to approximate the SDF as a simple linear mix of a small number of traded factors.

Chen, Pelger, and Zhu changed things a lot with their Deep Learning SDF approach. They argued that the real SDF is a very non-linear function that depends on a huge amount of information – things like overall economic conditions and details specific to each firm. Their setup uses a few key pieces:

(1) Feed-forward Networks: These handle the complicated non-linear links among hundreds of firm-level characteristics.

(2) LSTM Networks: They pull out changing macroeconomic conditions from time-series data, because risk premia are not constant – they shift over time.

(3) Generative Adversarial Networks (GANs): The GAN part helps find the assets that are toughest to price correctly. Instead of just trying to predict returns, the whole model works to minimize pricing errors that would allow arbitrage, sticking to the no-arbitrage rule [2].

The results stand out. Out-of-sample, their Deep Learning SDF gets a Sharpe ratio of 2.80 – more than twice that of standard linear models. The result explains much more of the differences in returns across stocks. Most importantly, it shows that the true pricing kernel relies on tricky interactions – for example, how value stocks perform very differently when inflation is high versus low. Linear models simply cannot capture that kind of complexity because of their math.

3.2. Autoencoders and deep portfolio theory

Portfolio optimization has its own issues. The classic Markowitz mean-variance method depends on estimating the covariance matrix of returns. When you deal with big portfolios like the S&P 500, small errors in that matrix make the whole optimization unreliable and extreme.

Heaton, Polson, and Witte came up with Deep Portfolio Theory using autoencoders. An autoencoder is a neural network that learns to reconstruct its input after squeezing it through a narrow hidden layer – the bottleneck. This forces it to find a compact, low-dimensional view (latent factors) of how the market really moves [1].

Unlike PCA, which only finds linear patterns, autoencoders pick up non-linear co-movements – things like how assets crash together in bad times. They suggest a four-step process: Encode the data to get latent factors, calibrate to spot anomalies, validate the findings, and verify. Stocks that have high reconstruction error—meaning the autoencoder cannot explain their returns well from the main market patterns—stand out as potentially disconnected from the crowd and could point to real opportunities. This method basically automates hunting for smart beta strategies without needing a human to decide what to look for.

DOI: 10.54254/2754-1169/2026.LD31802

4. Architectures for financial time series forecasting

4.1. Evolution of recurrent neural networks

Predicting financial time series is harder than many other kinds because the data is non-stationary—the statistical properties keep shifting—and volatility tends to come in bursts or clusters. Standard ARIMA models just aren't built for that and usually do poorly.

The first serious deep learning approach was Recurrent Neural Networks (RNNs). RNNs read the data one time step after another and carry a hidden state that is supposed to remember earlier information. The catch is that plain RNNs run into the vanishing gradient problem: gradients shrink so much during training that the network forgets what happened far back in the sequence.

To get around this, people started using Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs). LSTMs add three gates—input gate, output gate, forget gate—that control exactly what information gets kept, discarded or passed forward. That makes it possible for the model to hang on to long-term patterns, like a trend that kicked off six months earlier. Plenty of studies show LSTMs beat ARIMA when it comes to forecasting stock prices or volatility. GRUs do roughly the same job in many cases but train faster and use less memory because they have fewer gates.

4.2. The transformer revolution and TFT

Even with LSTMs, the sequential processing means the model still looks at things one step at a time, which makes it hard to spot broad or global patterns across the whole history. The Transformer, first built for language work, dropped the recurrence altogether and introduced self-attention. Self-attention lets the model look at every past time step all at once and decide on the fly which ones are most relevant right now.

Lim et al. [3] took that idea and created the Temporal Fusion Transformer (TFT) tailored for financial forecasting. TFT is made for multi-step ahead predictions and tries to be more interpretable than most deep models. It brings in a few important pieces: (1) Gated Residual Networks (GRN): These automatically dial down or shut off inputs that look like noise. (2) Variable Selection Networks: At each point in time, the model figures out which features matter most – maybe volume today, momentum yesterday, etc. (3) Static Covariate Encoders: These let the model use fixed information like sector ("Tech" or "Energy") to adjust the forecast and share learning across different stocks [3].

On standard test sets, TFT beats older baselines like DeepAR and regular LSTMs. One of the best things is the attention weights: you can actually see which parts of the past the model is paying attention to, helping traders spot different market regimes or key events the model is relying on.

On top of that, Graph Neural Networks (GNNs) have started to appear for handling how stocks connect to each other. Li et al. look at combinations of LSTM and GNN where the GNN learns the "spatial" links—things like which companies supply each other or are in the same supply chain – while the LSTM still handles the time side. By passing information through the graph of relationships, these models catch spillover effects and shocks moving from one stock to another, which plain single-series models completely ignore and that cuts down forecast errors [4].

DOI: 10.54254/2754-1169/2026.LD31802

5. Natural Language Processing (NLP) and alternative data

5.1. FinBERT: domain-specific pre-training

According to the Efficient Market Hypothesis, asset prices should already bake in every piece of available information. A huge chunk of that information sits in unstructured text—news articles, company filings, earnings call transcripts, and so on. The problem is that off-the-shelf NLP models trained on everyday English don't handle financial language well. Words like "liability" carry a negative tone in normal text but are just neutral descriptions in balance sheets or accounting reports.

Araci tackled this head-on with FinBERT. It's built on the BERT architecture but gets extra pre-training specifically on financial documents. The training corpus includes big collections like Reuters TRC2 and the Financial PhraseBank. FinBERT goes through the usual BERT tasks—Masked Language Modelling (where random words are hidden and the model guesses them) and Next Sentence Prediction (figuring out if two sentences logically follow each other)—but now it learns the special patterns, jargon and context that show up in finance [5].

When tested, FinBERT does much better than standard BERT or older models like LSTM on financial sentiment classification. People have built trading strategies that use FinBERT sentiment scores from news, and they end up with noticeably higher Sharpe ratios. That suggests the model picks up subtle signals in text that the market hasn't fully reacted to yet, giving a real edge before prices adjust.

5.2. Large Language Models (LLMs) in finance

Things took another jump with the arrival of powerful generative models like GPT-4 and LLaMA. A survey by Misra et al. in 2024 looks at how these LLMs are being used for financial reasoning and pulling information out of documents. Unlike older discriminative models such as FinBERT (which mainly classify or tag text), LLMs can do much more open-ended work. They can read through a full 10-K filing, spot and extract specific tables, work out financial ratios, highlight key risks and write a clear summary—basically acting like a junior analyst who never sleeps [6].

Some recent setups, including entries in the FinRL-Contest, feed LLM-generated insights from news straight into reinforcement learning agents to decide trades. That combination has shown promise for creating signals.

Still, there are real drawbacks. LLMs sometimes "hallucinate"—they confidently make up numbers or facts that aren't there. Another worry is that most testing happens in normal market conditions; there is not enough evidence on how the market behaves during extreme events like the 2008 crash or other tail-risk periods. That leaves questions about whether they would hold up when it really matters.

6. Algorithmic trading and reinforcement learning

6.1. The FinRL framework

Trading involves making decisions one after another as the market moves, so reinforcement learning (RL) suits it perfectly. In RL, an agent sees the state (current market data), chooses an action (buy, sell, or hold), and gets a reward (usually profit or loss adjusted for risk).

Liu et al. created FinRL, an open-source library that standardizes deep reinforcement learning for finance. It helps solve the reproducibility problem where different papers use different setups. FinRL includes: (1) Market Environments: Gym-style simulators for stocks, crypto, or portfolio

DOI: 10.54254/2754-1169/2026.LD31802

management. (2) Agents: Code for PPO, DDPG, SAC, and other key RL algorithms. (3) DataOps: Automatic pipelines to clean data and adjust for splits, dividends, etc [7].

FinRL also supports ensemble strategies. You run multiple sub-agents (one for trends, one for mean reversion, etc.) and a supervisor agent picks which to follow depending on current conditions. Tests show this beats simple benchmarks like buy-and-hold or minimum-variance portfolios.

6.2. High-frequency trading and market making

In high frequency trading, the main activity for many is market making: quoting buy and sell prices to provide liquidity, earn the spread, and avoid getting stuck with bad inventory. Traditional models like Avellaneda-Stoikov assume prices follow strict diffusion processes.

RL provides a model-free option. The agent learns optimal bid/ask quotes by interacting with a limit order book simulator. The rewards are spread earned but punish large inventory imbalances, so it learns risk-averse behavior. Studies show RL agents develop smart tactics, like adjusting quotes to reduce positions before expected moves, and often beat classical models in realistic tests.

7. Challenges: explainability and robustness

7.1. The "black box" problem and XAI

Deep learning models give answers but don't tell you why. This is a real problem in finance. Rules like SR 11-7 say every model has to be auditable – you must explain every decision clearly.

Explainable AI (XAI) is the attempt to fix it [8]. Wilson puts the main ways into three groups for finance: (1) Ante-hoc – models that are easy to understand right from the start. A decision tree is a good example. You can look at the splits and see what they mean in business terms. (2) Post-hoc – ways to explain complicated models after they run. SHAP and LIME are used a lot. SHAP shows how much each input changed the answer, like "P/E ratio added +0.5% to the return prediction". (3) Counterfactuals – simple what-if questions. For example, "If volatility was 5% higher the model would have sold".

XAI is needed so regulators and risk teams can check that the model isn't using wrong patterns or forbidden data that could cause bias.

7.2. Synthetic data and GANs

Models trained on past real data often fit the history too well. When markets change fast – like in COVID-19 – they break. Baviskar calls this "Conditional Adversarial Fragility". The model works fine normally but falls apart with even small stress [9].

Researchers use GANs to make fake data. TimeGAN learns how real prices move over time and creates new price series that look real but never happened [10].

8. Conclusion

Putting machine learning to work in quantitative finance has truly shaken things up in how investments are managed. It became clear that these new approaches often deliver real advantages over the older statistical models that have been around for decades. Take asset pricing, for instance. Models built on deep learning, such as the ones put forward by Chen and Pelger, do a better job at spotting those tricky non-linear risk factors. Traditional linear models simply pass them by, so this leads to sharper estimates of what returns we might expect.

DOI: 10.54254/2754-1169/2026.LD31802

Then there are the forecasting tools. Things like Temporal Fusion Transformers—often shortened to TFT—and Graph Neural Networks have raised the bar quite a bit. They handle long-term patterns in data and map out how different assets link together, resulting in predictions that tend to hold up better. On the language side, models like FinBERT have turned messy text from news or central bank reports into solid measures of market sentiment. That adds a layer of insight that was hard to get before. And in trading, reinforcement learning setups, including frameworks like FinRL, are pushing automated systems to adapt in real time, rather than sticking to fixed rules that can fall apart when markets shift.

All the same, people in the field are starting to look beyond raw performance metrics. Researchers and practitioners are increasingly emphasizing the need for trustworthy models in quantitative finance. Greater attention is now being directed towards Causal AI, which seeks to identify the underlying causes of price movements rather than merely detecting statistical patterns or correlations. This shift reflects a broader recognition that understanding genuine causal mechanisms is essential for building robust, reliable, and regulatorily compliant systems in real-world financial applications. It also involves building systems that hold up better when markets turn unpredictable.

With computing power still climbing and techniques for explaining AI decisions getting sharper, the old split between classic quant methods and AI-driven ones should start to blur. What we'll probably end up with is a single approach that pulls together sound economic principles with clever computational tricks to create something more reliable and transparent.

References

[1] Heaton, J. B., Polson, N. G., & Witte, J. H. (2016). Deep portfolio theory. arXiv preprint arXiv: 1605.07230. https: //arxiv.org/abs/1605.07230

[2] Chen, L., Pelger, M., & Zhu, J. (2020). Deep learning in asset pricing. arXiv preprint arXiv: 1904.00745. https: //arxiv.org/abs/1904.00745

[3] Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. International Journal of Forecasting, 37(4), 1748–1764. https: //doi.org/10.1016/j.ijforecast.2021.03.012

[4] Li, Y., Pan, Y., & Yang, X. (2024). A systematic review on graph neural network-based methods for stock market forecasting. ACM Computing Surveys. https: //dl.acm.org/doi/abs/10.1145/3638243

[5] Araci, D. (2019). FinBERT: Financial sentiment analysis with pre-trained language models. arXiv preprint arXiv: 1908.10063. https: //arxiv.org/abs/1908.10063

[6] Misra, S., et al. (2024). Language modeling for the future of finance: A quantitative survey. arXiv preprint arXiv: 2402.12262. https: //arxiv.org/abs/2402.12262

[7] Liu, X. Y., Yang, H., Chen, Q., Zhang, R., Yang, L., Xiao, B., & Wang, C. D. (2020). FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance. Deep Reinforcement Learning Workshop, NeurIPS 2020. https: //arxiv.org/abs/2011.09607

[8] Wilson, D. (2024). Explainable AI in finance. CFA Institute Research Reports. https: //rpc.cfainstitute.org/en/research/foundation/2024/explainable-ai-in-finance

[9] Baviskar, S. (2025). Conditional adversarial fragility in financial machine learning under macroeconomic stress. arXiv preprint arXiv: 2512.19935. https: //arxiv.org/abs/2512.19935

[10] Yoon, J., Jarrett, D., & van der Schaar, M. (2019). Time-series generative adversarial networks. Advances in Neural Information Processing Systems, 32. https: //papers.nips.cc/paper/8789-time-series-generative-adversarial-networks