Theory and Practice of Science and Technology
2025, VOL. 6, NO. 10, 93-95
DOI: 10.47297/taposatWSP2633-456918.20250610

A Review of Machine Learning and Automated Trading for Foreign-Exchange Trend Forecasting

Jiaqi Zhao

University of Birmingham, Birmingham, West Midlands, B15 2TT, United Kingdom

ABSTRACT
Foreign-exchange (FX) markets are liquid, data-rich, and highly non-stationary. Classic evidence shows that, at common horizons, macro and time-series models often fail to beat a naive random walk, motivating a shift toward flexible machine-learning (ML) methods alongside stricter evaluation protocols [1]. This review synthesizes recent progress in ML for FX trend prediction and the practical pipeline that connects forecasts to execution on MetaTrader 5 (MT5). I cover features drawn from prices, volatility, microstructure, and text; compare statistical baselines, tree-boosting, recurrent networks, and transformer architectures; and emphasize time-aware validation, forecast-comparison tests, and overfitting controls. Finally, I discuss implementation patterns for turning Python models into automated trades on MT5 via the official Python module and custom-symbol tooling. The goal is to map an end-to-end, evaluation-first approach that yields empirically credible and operationally viable FX trend models [1-2].

KEYWORDS
Foreign exchange; Machine learning; Trend forecasting; Automated trading; MetaTrader 5

1 Introduction

The empirical literature on exchange-rate forecasting famously documents that, over horizons from one to twelve months, the out-of-sample accuracy of structural and time-series models seldom exceeds a random walk; this "Meese–Rogoff puzzle" reframed expectations about what constitutes a meaningful edge and pushed researchers toward methods that can adapt to non-stationarity while being evaluated with walk-forward protocols [1]. In parallel, microstructure research argued that order flow—the signed volume of trades—acts as the proximate driver of short-horizon price changes, implying that high-frequency signals may emerge from limit-order-book (LOB) dynamics rather than from macro series alone [3]. These perspectives, combined with advances in representation learning and scalable compute, set the stage for FX-specific applications of tree-boosting, recurrent networks, and transformers, as well as sentiment models tuned to financial text such as FinBERT [4].

2 Data and Feature Landscape

A practical FX pipeline begins with price and returns series at multiple horizons, where basic differencing and rolling statistics supply the initial directional features, but evaluation remains anchored to a random-walk baseline to calibrate difficulty [1]. Because return variance is time-varying, many workflows add volatility states from GARCH models or realized-vol proxies, either to normalize returns or to condition signals on regimes [2]. When high-granularity feeds are available, researchers enrich the feature space with microstructure descriptors such as order-flow imbalance, queue dynamics, and LOB depth; sequence models trained on such event streams often capture short-horizon predictability [3,5]. Text adds another orthogonal channel: transformer models specialized for finance—most notably FinBERT—enable sentiment factors from headlines, reports, and policy statements that can complement purely price-based features in medium-frequency settings [4,6]. Finally, platform realities matter: on MT5, the list and naming of tradable symbols is broker-specific; unavailable pairs can be added as custom symbols with imported bars or ticks, which is essential for reproducible research and for aligning paper-to-practice [7].

3 Methods: From Statistical Baselines to Deep Sequence Models

Method selection follows from the data's granularity and the horizon of interest. Classical ARIMA and exponential-smoothing remain the first yardsticks; they encode interpretable dynamics and help distinguish "easy" seasonal structure from genuine predictive power, while GARCH family models describe conditional variance and often feed into risk-aware signal scaling [8,2]. On tabular, multi-source feature sets—think rolling returns, volatility states, calendar effects, and sentiment scores—gradient-boosted trees such as XGBoost provide strong non-linear baselines with fast training and clear feature attributions [9]. When sequential dependencies dominate, recurrent networks (LSTM/GRU) and 1-D CNNs offer compact inductive biases for intraday data; at longer horizons or with heterogeneous covariates, transformer families like the Temporal Fusion Transformer (TFT) and Informer introduce attention mechanisms, multi-horizon outputs, and built-in variable selection that can be useful for mixing static descriptors (e.g., country), known-future covariates

(calendar), and observed history (prices/volatility/news) [10-11]. Finally, for environments with rich microstructure records, deep models trained directly on LOB events have repeatedly demonstrated short-horizon gains by exploiting order-flow dynamics [5]. The common theme is to benchmark against the random walk, add complexity only when justified out-of-sample, and keep models small enough to survive regime changes.

4 Evaluation: Time-Aware Validation and Forecast Comparison

Evaluation is where most FX studies win or lose credibility. Random K-fold cross-validation is inappropriate for time-ordered data because it leaks future information; instead, one should use expanding or rolling windows and purged gaps between train and test segments when serial dependence is high. In scikit-learn this is operationalized via TimeSeriesSplit, and practical variants can group by session or asset when needed to avoid cross-contamination [12-13]. Beyond raw errors (MAE/MSE), differences in accuracy need statistical testing: the Diebold–Mariano (DM) test evaluates whether two forecast loss series differ significantly, and the Giacomini–White framework extends this to conditional predictive ability, which is crucial under heteroskedasticity and time-varying regimes [14-15]. For trading strategies, one should report distributional metrics across walk-forward slices rather than a single headline number; confidence intervals on annualized returns and Sharpe, with explicit accounting for costs and slippage, make results interpretable and resistant to cherry-picking.

5 From Prediction to Execution on MT5

Translating scores into trades proceeds in two layers: a research backtester and a live-execution bridge. Python backtesting frameworks such as backtrader and Zipline let you encode entry/exit logic, slippage, and commissions while iterating quickly on signals and position-sizing rules; they are event-driven and support multiple data feeds [16-17]. For MT5 deployment, the official MetaTrader5 Python package connects to a running terminal for data and order routing (e.g., initialize, copy_rates_from_pos), while MQL5 Expert Advisors (EAs) can receive signals by calling Python or via inter-process messaging such as ZeroMQ [22]. The exact choice depends on latency and maintainability: direct Python calls are simpler; socket bridges are more flexible in multi-process setups. Broker heterogeneity complicates reproducibility, so production code should enumerate available symbols, resolve suffix/prefix variants, and, where necessary, create custom symbols and import bars from CSV/JSON to keep pairs consistent across brokers [18,7].

6 Comparative Evidence: What Works Where

At daily and monthly horizons, the random-walk benchmark remains hard to beat on average, which is why many successful applications concentrate either on high-frequency microstructure (where order flow and LOB context provide short-lived edges) or on medium-frequency fusions that combine technical, calendar, and sentiment signals to exploit episodic predictability [1,3]. Gradient-boosted trees are strong in the latter setting because they handle mixed tabular features and yield interpretable importances, while LSTMs/GRUs and compact transformers shine when sequential dependencies and known-future covariates matter; TFT's variable-selection and attention blocks have been leveraged to diagnose regime-relevant drivers even when overall gains are modest [9-10]. Informer extends the transformer toolkit to very long histories with sub-quadratic attention, enabling experiments that would otherwise be memory-bound [11]. In tick-level studies, deep networks trained directly on LOB states have demonstrated directional accuracy improvements over standard classifiers, albeit with data and infrastructure requirements that exceed typical retail setups [5]. The emergent pattern is that horizon and data richness dictate the model: microseconds-to-minutes favor LOB-aware deep sequence models; hours-to-days often reward modest, robust tabular learners augmented by sentiment and volatility conditioning.

7 Overfitting Control and Robustness in Live-Trading Contexts

Backtest-overfitting is a structural risk whenever researchers iterate on features, windows, thresholds, and models. Practical defenses include nested or rolling cross-validation with a final untouched audit window; explicit reporting of the number of trials; and adjustment of summary statistics for multiple testing. The Deflated Sharpe Ratio (DSR) offers a principled way to correct performance for selection bias and non-normality, while the Probability of Backtest Overfitting (PBO) quantifies how likely a selected strategy is to disappoint out of sample [20-21]. These tools complement forecast-comparison tests such as DM and CPA and, together with cost-inclusive backtests, produce results that are less sensitive to researcher degrees of freedom and more indicative of real-money feasibility [14-15]. In the execution stack, robustness also demands realistic assumptions about spreads, commissions, and slippage, preferably stress-tested by sampling trade sequences or by replaying randomized micro-shocks to the order-arrival process.

8 Practical Recipes for an End-to-End Pipeline

Putting the components together, a pragmatic workflow starts with clean OHLCV extracted from MT5 or institutional

feeds, normalized for time zones and trading calendars, and—with broker symbol differences in mind—augmented via custom symbols when instruments are unavailable; MT5 exposes APIs to create, populate, and manage such symbols from CSV or programmatically [19]. Features should be introduced incrementally: a lean baseline of returns, rolling volatility, and calendar dummies establishes the reference; volatility-aware scaling (via GARCH states) and sentiment (FinBERT on major headlines) can then be added if they improve out-of-sample risk-adjusted returns [2,4]. Models advance from random walk and ARIMA baselines to XGBoost and, when justified by sequential dependencies or heterogeneous covariates, to compact LSTMs or TFT; each step is evaluated with walk-forward splits (TimeSeriesSplit), costs and slippage, and DM/CPA tests on error series [12,14-15]. Deployment proceeds either by issuing orders from Python through the MetaTrader5 package or by sending signals to an MQL5 EA via sockets (e.g., ZeroMQ bindings) [18,22]; in both cases, symbol resolution, connectivity checks, throttling, and logging are non-optional for reliability (MQL5 Python Integration; ZeroMQ bridge references). Iterations close the loop by folding live telemetry—fills, rejects, latencies—back into feature and risk-control design.

9 Conclusion

FX trend forecasting is hard in expectation and unforgiving in evaluation. The random-walk baseline remains formidable at daily-plus horizons; microstructure-informed deep models and sentiment-augmented tabular learners offer credible edges in specific regimes, but only when matched to horizon and validated under walk-forward protocols with formal forecast-comparison tests [1,14]. On the engineering side, MT5 provides a practical bridge from Python research to live execution, provided one handles broker-specific symbol catalogs and exploits custom-symbol tooling to keep data consistent [7]. Ultimately, the most durable results come from an evaluation-first mindset: lean baselines, time-aware validation, cost-inclusive backtests, and explicit overfitting controls [20-21].

References

[1] Meese R A, Rogoff K. Empirical Exchange Rate Models of the Seventies: Do They Fit Out of Sample [J]. Journal of International Economics, 1983, 14(1-2): 3-24.

[2] Bollerslev T. Generalized Autoregressive Conditional Heteroskedasticity [J]. Journal of Econometrics, 1986, 31(3): 307-327.

[3] Evans M D D, Lyons R K. Order Flow and Exchange Rate Dynamics [J]. Journal of Political Economy, 2002, 110(1): 170-210.

[4] Araci D. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models [J/OL]. arXiv, 2019, 1908.10063.

[5] Sirignano J. Deep Learning for Limit Order Books [J]. Quantitative Finance, 2019, 19(4): 549-570.

[6] Jiang T, et al. Financial Sentiment Analysis using FinBERT with LSTM [J/OL]. arXiv, 2023, 2306.02136.

[7] MetaQuotes Ltd. MQL5 Documentation: Symbols List and Custom Symbols [EB/OL]. https://www. mql5. com/en/docs/customsymbols, [accessed 2025-10-17].

[8] Box G E P, Jenkins G M, Reinsel G C, Ljung G M. Time Series Analysis: Forecasting and Control [M]. Hoboken: Wiley, 2015.

[9] Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System [C]// Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2016: 785-794.

[10] Lim B, Arik S Ö, Loeff N, Pfister T. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting [J]. International Journal of Forecasting, 2021, 37(4): 1748-1764.

[11] Zhou H, Zhang S, Peng J, et al. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting [C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2021.

[12] Pedregosa F, et al. Scikit-learn: Machine Learning in Python [J]. Journal of Machine Learning Research, 2011, 12: 2825-2830.

[13] Raschka S. MLxtend: Providing Machine Learning Extensions to Python's Scientific Computing Stack [J]. Journal of Open Source Software, 2020, 5(46): 1956.

[14] Diebold F X, Mariano R S. Comparing Predictive Accuracy [J]. Journal of Business and Economic Statistics, 1995, 13(3): 253-263.

[15] Giacomini R, White H. Tests of Conditional Predictive Ability [J]. Econometrica, 2006, 74(6): 1545-1578.

[16] Backtrader. Backtrader Documentation and Quickstart Guide [EB/OL]. https://www. backtrader. com/docu/quickstart/quickstart/, [accessed 2025-10-17].

[17] Zipline. Zipline Documentation [EB/OL]. https://zipline.ml4trading.io/, [accessed 2025-10-17].

[18] MetaQuotes Ltd. MetaTrader5 Python Module Documentation (e. g., initialize, copy_rates_from_pos) [EB/OL]. https://www. mql5. com/en/docs/python_metatrader5/, [accessed 2025-10-17].

[19] MetaQuotes Ltd. MQL5: CustomRatesUpdate and Custom Symbols API [EB/OL]. https://www. mql5. com/en/docs/customsymbols/customratesupdate, [accessed 2025-10-17].

[20] Bailey D H, López de Prado M. The Deflated Sharpe Ratio: Correcting for Selection Bias, Backtest Overfitting and Non-Normality [J]. Journal of Portfolio Management, 2014, 40(5): 94-107.

[21] Bailey D H, Borwein J, López de Prado M, Zhu Q J. The Probability of Backtest Overfitting [J]. Journal of Computational Finance, 2017, 20(4): 39-69.

[22] Ding M. MQL-ZMQ: ZeroMQ for MQL [EB/OL]. GitHub repository. https://github.com/dingmaotu/mql-zmq, [accessed 2025-10-17].