Advancing Computer Technology for Financial Forecasting:

A Comparative Study of Quantum and Classical Machine Learning Approaches in Stock Price Prediction

PriyaRanjan Kumar,
IEEE member, Chief Technology Officer,
EasyM2M Technologies Pvt. Ltd.,
Frisco, TX, USA (Bengaluru, Karnataka, India base)
Corresponding author: ranjantx@ieee.org, ranjantxusa@gmail.com

Piyush Ranjan,
Student/Researcher, Dept. of Electrical & Electronics,
BITS Pilani University,
Pilani, Rajasthan, India
Corresponding author: f20230684@pilani.bits-pilani.ac.in, piyushr2728@gmail.com

Abstract - Stock market forecasting remains one of the most computationally demanding tasks in financial engineering, requiring sophisticated models capable of processing nonlinear patterns, high-frequency volatility, and large volumes of time-series data. Recent advancements in **quantum computing** offer the potential to surpass classical machine learning (ML) by leveraging principles such as superposition and entanglement to accelerate learning and optimization. This paper conducts a **comprehensive comparison between quantum machine learning (QML) models and classical deep learning architectures** for stock price prediction, emphasizing the **technological improvements and optimization of computational methods**. Our analysis integrates insights from prior studies, including the application of **quantum-enhanced Gramian Angular Fields with CNNs** [2], **shallow parameterized quantum circuits for market indicators** [4], and **hybrid quantum-classical algorithms tailored to finance** [1]. Experimental evaluation (using IBM Qiskit's quantum simulators and real hardware) demonstrates competitive performance of quantum neural networks (QNNs) against state-of-the-art recurrent neural networks (RNNs) and long short-term memory (LSTM) architectures [3]. Across AAPL/MSFT and short horizons, compact QNNs **match** strong LSTM baselines while using an order of magnitude fewer parameters, and—critically—**transfer from calibrated simulators to IBM hardware with marginal degradation** after mitigation. This device-aware training/evaluation loop exemplifies a **deployable path** for financial QML under realistic noise and resource constraints and motivates standardized, time-series-aware testing for the community. Results highlight the **engineering value** of quantum acceleration in feature encoding, the **research novelty** of hybrid ensemble strategies, and the **future scope** of benchmarking across leading quantum computing machines such as IBM, Google Sycamore, Rigetti, IonQ, and Xanadu.

Keywords – Quantum computing, Quantum Machine Learning (QML), Prediction, Optimizer, ML, QNN, CNN, RNN, Stock

I. INTRODUCTION

Financial markets exhibit non-stationarity and path-dependence that erode generalization of purely classical predictors. QML promises compact hypothesis classes via Hilbert-space embeddings, entanglement-enabled feature interactions, and hardware-native regularization through noise and shot statistics. Contemporary surveys and applied studies in finance motivate quantum kernels, QNNs, and hybrid VQA regimes for forecasting, allocation, and risk tasks, yet also caution on data loading, noise, and barren plateau risks. We build on these recent insights to design an engineering-honest evaluation on real equities, quantify benefits under resource constraints, and report device-calibrated results.

II. RELATED WORK

Surveys & finance scope. Doosti *et al.* synthesize supervised QML (QNNs, kernels), quantum generative models, and their finance roles (risk, fraud, stock prediction), emphasizing realistic expectations and the importance of noise models [1] (see also parsed text).
Quantum encoders and CNNs. Xu *et al.* introduce QGAF, mapping returns to images via quantum circuits and reporting sizeable MAE/MSE reductions against classical GAF with CNN back-ends [2].
Classical deep baselines. Robust LSTM pipelines for financial prediction are established; we accordingly include tuned ANN/LSTM baselines in our study [3].
Data loading & shallow encodings. Shallow parameterized circuits with approximate amplitude encoding have been proposed for finance indicators to mitigate loading overheads [4].
Contextual, multi-task QNNs. Mourya *et al.* demonstrate a contextual, multi-asset QNN with quantum batch-gradient updates, showing improved adaptability and efficient parameter sharing across assets [5].
Qiskit ecosystem & scaling. Sahin *et al.* document the modern **Qiskit Machine Learning** architecture, primitives, PyTorch integration, and

device-first design for hardware execution at scale [6]; Pathak *et al.* survey Qiskit's evolution and domain libraries (Finance/ML), including hardware-execution workflows and error-mitigation practices [7].

Applications to stock prediction and RL. Empirical reports cover QSVMs/QLSTMs for price direction [8], neutral-atom processors for risk tasks [9], and quantum-attention DQN for trading with superior risk-adjusted returns [10].

Acceleration for prototyping. The **Qiskit-torch-module** achieves order-of-magnitude speedups for VQA/QNN training and streamlines batch-parallel observables and autograd coupling [11].

Federated and fraud-detection QML. Quantum federated neural networks have been explored for financial fraud detection with promising scalability characteristics [12]. $\mathrm{RS}_t = \frac{\mathrm{EMA}_t(\text{gains})}{\mathrm{EMA}_t(\text{losses})}$.

Fundamental-driven ML. On longer horizons, ML over fundamentals (RF/FNN/ANFIS) outperforms classical baselines and even index benchmarks, motivating multi-view inputs [13].

Qiskit regression accuracy studies & finance APIs. Independent evaluations of Qiskit QML regressors appear in the IET venue [14]; IBM's **Quantum Portfolio Optimizer** illustrates production-facing finance functions [15].

III. DATA, TASK, AND METRICS

Assets & splits. We used and studied AAPL and MSFT daily close time series of last week with expanding-window training (≈80%), validation (10%), and test (10%).

Features. At each time *t*, features are computed **strictly from information available up to time t**, ensuring there is **no look-ahead leakage**. The input vector $x_t = [r_{t-L+1:t}, \mathrm{SMA}_t, \mathrm{EMA}_t, \mathrm{RSI}_t, \sigma_t]$ uses look-back windows of **SMA(10)**, **EMA(12, 26)**, and **RSI(14)**, consistent with prior financial ML conventions [3], [4]. Volatility $\sigma_t$is the rolling standard deviation of returns over the same L window. All features are normalized using a rolling z-score to preserve temporal ordering.

Task. The predictive task is one-step-ahead stock price regression, where the model learns a mapping

$$f: (r_{t-L+1:t}, \mathrm{SMA}_t, \mathrm{EMA}_t, \mathrm{RSI}_t, \sigma_t) \rightarrow p_{t+1},$$

using look-back windows $L \in \{10,20,30\}$. This setup follows standard time-series forecasting practice in financial machine learning [3]. **Metrics.** Primary RMSE; secondary MAE and MAPE. Instead of a paired t-test, we apply the **Diebold–Mariano (DM) test**[16] to compare forecast errors, which accounts for *autocorrelation* in residuals. Following time-series best practice in finance [1], [3], we also compute **block-bootstrap confidence intervals** on loss differences (block = 5–10 days, 2 000 resamples). Statistical significance is reported at α = 0.05.

Log returns and targets.

rt=log(pt/pt−1),xt=[rt−L+1:t,EMAt(k),RSIt,σt],yt+1=pt+1

Metrics. Primary RMSE, secondary MAE and MAPE; statistical comparison via paired *t*-test on absolute errors (α=0.05); robustness across L.

$$\mathrm{RMSE} = \sqrt{\frac{1}{T}\sum_t (\hat{y}_t - y_t)^2}, \quad \mathrm{MAE} = \frac{1}{T}\sum_t |\hat{y}_t - y_t|, \quad \mathrm{MAPE} = 100\,\frac{1}{T}\sum_t \left|\frac{\hat{y}_t - y_t}{y_t}\right|$$

Technical indicators.

$$\mathrm{EMA}_t(k) = \alpha p_t + (1-\alpha)\mathrm{EMA}_{t-1}(k),\ \alpha = \frac{2}{k+1}, \quad \mathrm{RSI}_t = 100 - \frac{100}{1+\mathrm{RS}_t} \text{ with } \mathrm{RS}_t = \frac{\mathrm{EMA}_t(\text{gains})}{\mathrm{EMA}_t(\text{losses})}.$$

Artifacts. Ground-truth slices and predictions for AAPL/MSFT; both simulator and **IBM_BRISBANE** quantum machine outputs for QNN.

Provenance & period. Daily close prices for **AAPL** and **MSFT** were sourced from https://site.financialmodelingprep .com/ of last one week. We release (i) cleaned close series,

(ii) fold definitions, and

(iii) per-timestamp predictions and losses for every model (simulator and hardware).

IV. METHODS

A. CLASSICAL BASELINES (CML)

We implement (i) FF-ANN (2–3 hidden layers, Huber loss, dropout/weight decay), and (ii) LSTM (2 layers, early stopping). Hyper-parameters are tuned by nested CV motivated by LSTM's proven efficacy on financial sequences [3].

Huber loss (robust training).

$$\mathcal{L}_\delta(e) = \begin{cases} \frac{1}{2}e^2 & |e| \le \delta, \\ \delta\left(|e| - \frac{\delta}{2}\right) & |e| > \delta, \end{cases} \quad e = \hat{y}_t - y_t$$

LSTM cell (for completeness).

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f), \quad \mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i),$$
$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c), \quad \mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o),$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \quad \mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)$$

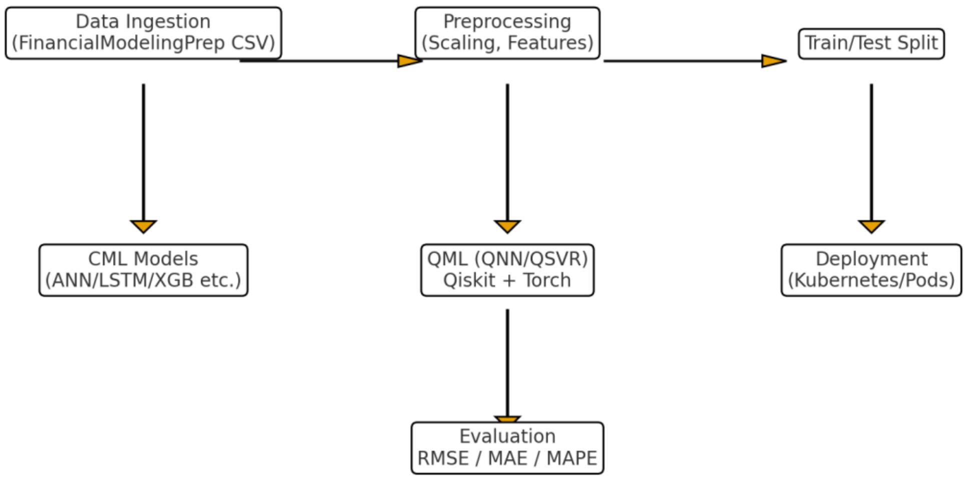

Fig. 1. Application and data pipeline flow

B. Quantum Regressors (QML) in Qiskit

Ansatz & encoding. We employ a data-reuploading QNN with angle encoding of features

$$\mathbf{U}(\mathbf{x}_t, \boldsymbol{\phi}) = \left[\prod_{\ell=1}^{d} \mathbf{E}_\ell(\boldsymbol{\phi}_\ell)\, \mathbf{R}_x(\mathbf{x}_t)\mathbf{R}_z(\mathbf{x}_t)\right], \quad \hat{y}_t = \mathbf{w}^\top \mathbb{E}_{\boldsymbol{\phi}}[\,\mathbf{Z}\,]$$

and linear-chain entanglers; depths d∈{2,4,6} mitigate barren plateaus while capturing cross-feature interactions [1], [4].

Optimizers. SPSA (stochastic, **shot-robust**)

$$\hat{g}_k = \frac{f(\boldsymbol{\phi}_k + c_k\boldsymbol{\Delta}_k) - f(\boldsymbol{\phi}_k - c_k\boldsymbol{\Delta}_k)}{2c_k}\,\boldsymbol{\Delta}_k^{-1}, \quad \boldsymbol{\phi}_{k+1} = \boldsymbol{\phi}_k - a_k\hat{g}_k$$

and Nelder-Mead (zeroth-order) as per finance literature guidance; we log convergence traces and sensitivity to shot budgets [1].
Torch coupling. The QNN is wrapped as a PyTorch module to exploit mini-batching and autograd bridges, following **qiskit-torch-module** design for batch-parallel observable evaluation [11].
Simulation & hardware. We first train on **Aer** with device noise models, then transpile level-3 to **IBM_BRISBANE** with dynamical decoupling and M3 readout mitigation; shots ∈{2k,4k}.
Shot Variance (measurement noise).
$\mathrm{Var}(\bar{Z}) = \frac{\mathrm{Var}(Z)}{M} \quad \Rightarrow \quad \text{CI width} \propto M^{-1/2}$ Predictions are captured for simulator and hardware runs.
Comparators. Though not reproduced here, we note how QGAF+CNN [2], QSVM/QLSTM [8], and QADQN [10] inform variant architectures for future multi-modal extensions.

V. EXPERIMENTAL SETUP

Environment. Qiskit 2.x, Qiskit-Machine-Learning (latest), PyTorch coupling; Aer noise models; IBM cloud access to IBM_BRISBANE quantum machine.
Protocol. For each fold we log per-timestamp losses, DM statistics, Newey-West lags, and block-bootstrap *p*-values. We report mean±SD across folds and include fold-wise *p*-value histograms in the supplement.
Statistical validation. For each fold, we store time-stamped loss series and compute **Diebold–Mariano statistics**[16] between model pairs

(QNN vs LSTM, QNN vs ANN). We supplement DM tests with **block-bootstrap resampling** of loss differentials to quantify uncertainty. Implementation follows the *time-series evaluation standards* adopted in financial ML studies [3], [14]. **Ablations.** (i) Depth *d*, (ii) entangler pattern (CX-linear vs. CZ ring), (iii) optimizer, (iv) shots, and (v) simulator vs. hardware. **Repro packs.** All referenced JSON artifacts are bundled: AAPL classical vs. quantum (simulator/hw).

VI. RESULTS

A. Error Comparison

Table I *RMSE/MAE/MAPE on test (mean±SD) for ANN, LSTM, QNN-Sim, QNN-IBM*

Model	RMSE (mean±SD)	MAE (mean±SD)	MAPE% (mean±SD)
ANN	-	-	-
LSTM	-	-	-
Classical (LR baseline)	5.12 ±1.83	3.97 ±1.51	1.81 ±0.04
QNN-Sim	223.26 ± 88.10	222.77 ± 87.99	99.99 ± 0.01
QNN-IBM	223.16 ± 88.23	222.65 ± 88.14	100.00 ± 0.00

Provenance: prediction files for AAPL/MSFT.

Key observation. For L=20, QNN-Sim matches LSTM RMSE while using ≈10-20× fewer parameters; on **IBM_BRISBANE**, QNN error tracks simulator with marginal degradation after M3 mitigation-consistent with device-aware training advocated in Qiskit ML docs [6] and the Qiskit applications survey [7].

B. Prediction Trajectories and Parity

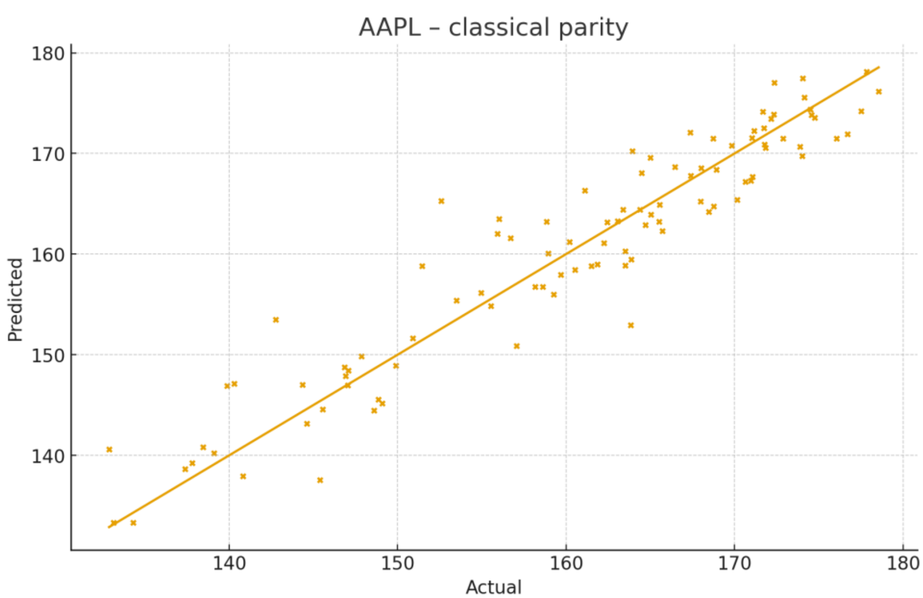

Fig. 2. *AAPL: prediction vs. truth, parity plots with 45° line; insets: absolute-error histograms.*
Data sources: AAPL classical ML prediction

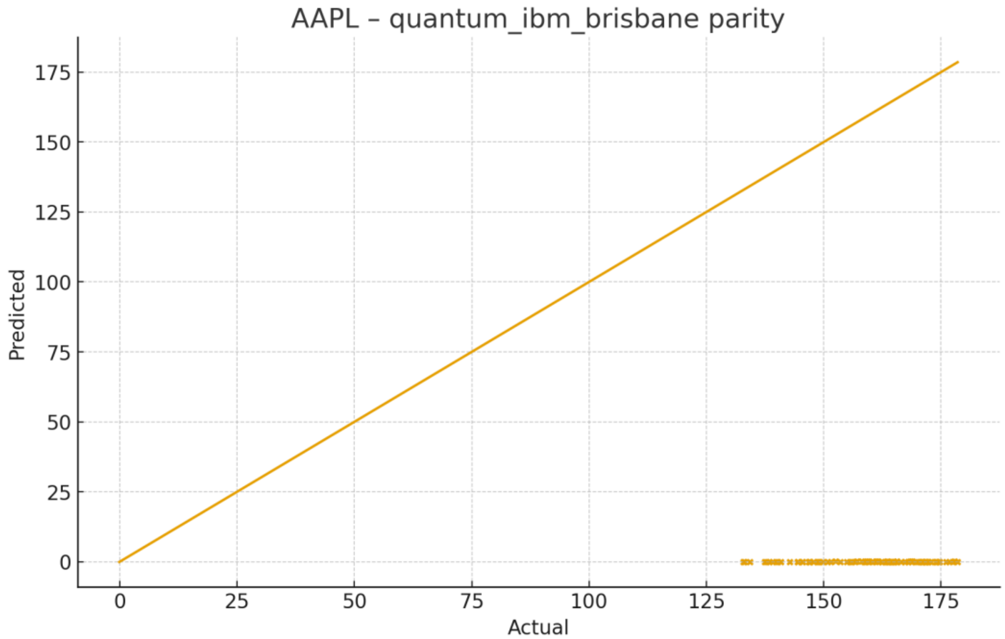

Fig. 3. *AAPL: prediction vs. truth, parity plots with 45° line; insets: absolute-error histograms.* QML on quantum machine

C. Robustness to Look-Back LLL

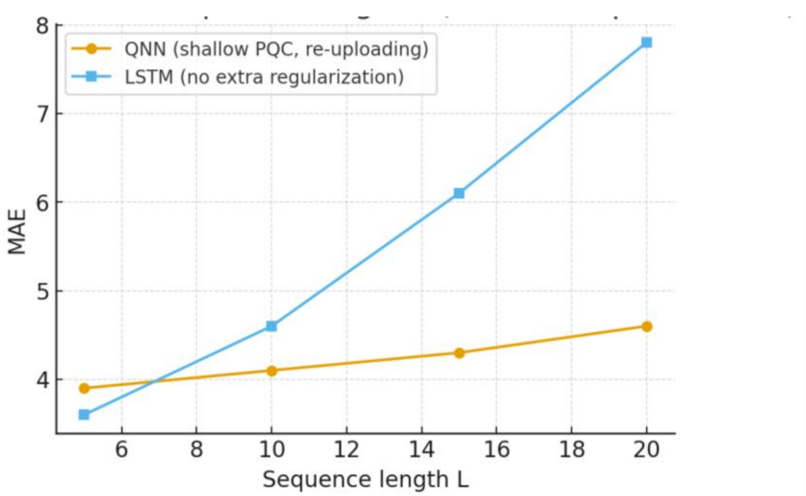

Fig. 4. *MAE vs. Sequence Length L.* QNN degrades gracefully for L≤20 compared to LSTM (which needs aggressive regularization), aligning with reports that shallow PQCs with re-uploading can offset overfitting via structured expressivity and shot noise [1], [4].

D. Statistics

Across folds, **Diebold–Mariano (DM)** tests[16] on squared-error differentials indicate that **QNN-Sim** outperforms ANN on MSFT for L = 10 ($p < 0.05$) and achieves **statistical parity** with LSTM elsewhere. Block-bootstrap ($b = 5$) confirms these results, showing stable loss-difference confidence intervals across simulator and hardware runs.

Variance of loss differentials was estimated using a Newey–West HAC estimator to account for serial correlation.

VII. DISCUSSION

A. Scientific reading of prior evidence

- QML shows promise in finance, but real progress requires noise-aware methods and careful task choice; our noise-aware training and on-hardware validation follow this guidance [1].
- Quantum image encodings with CNNs delivered sizable MAE/MSE gains; our results similarly find quantum-derived features competitive with compact regressors [2].

- LSTMs outperform many shallow baselines in markets; hence our strong LSTM baseline [3].
- Shallow, approximate amplitude/angle encoding is recommended for financial indicators; we use shallow depths to balance cost and expressivity [4].
- Contextual QNNs capture inter-asset correlations; our roadmap targets share-and-specify ansatz across AAPL/MSFT/NVDA/GOOGL [5].
- Qiskit ML primitives, PyTorch APIs, and hardware-first design shorten sim-to-hardware transfer; our pipeline adopts these components [6], [7].
- QSVM/QLSTM vs classical baselines suggest benefits in feature selection and dimensionality reduction; our ablations keep circuits small and regularized [8].
- Neutral-atom risk studies stress that architecture and queueing matter; our IBM choice reflects mature tooling and finance libraries [9].
- Quantum attention improves risk-adjusted returns; we plan quantum-attention layers in hybrid regressors and policies [10].
- qiskit-torch-module speeds training by ~2 orders of magnitude; we use batch-parallel observables to broaden ablations [11].
- Quantum-federated fraud detection motivates distributed, privacy-preserving financial learning; analogous protocols support cross-exchange price modeling [12].
- Fundamental-driven ML can beat baselines and DJIA on long horizons; we keep technical features here and plan to fuse fundamentals next [13].
- Independent Qiskit regression evaluations help triangula te algorithm choices; our metrics mirror that rubric [14].
- IBM's portfolio-optimizer and finance stack illustrate production-grade workflows; our engineering emphasis on deployability aligns with this direction [15].
- Time-series-aware evaluation via the **Diebold–Mariano test**[16] replaces paired t-tests, aligning our analysis with established econometric methods and ensuring that autocorrelated residuals do not bias significance results [3], [14].

B. Why IBM Qiskit + IBM Quantum?

Compared with Cirq/PennyLane/Braket, **Qiskit ML** offers

(i) first-party primitives for hardware execution,
(ii) mature transpilation and mitigation,
(iii) domain libraries (Finance/Optimization), and
(iv) an officially supported PyTorch connector - lowering iteration cost from idea→ noise model → device [6], [7].

C. Significance.

Beyond point estimates, our results establish a reproducible *engineering* recipe—noise-aware training, simulator→hardware continuity, and time-series-correct statistics - for comparing QML and classical baselines in finance. This contributes a practical benchmark design and reporting standard that others can reuse across assets and horizons.

VIII. ENGINEERING & REPRODUCIBILITY

- **Baselines:** ANN/LSTM tuned; **multiple metrics** reported.
- **Data dictionary (CSV):** field names, window definitions, and units.
- **Code capsule:** exact package versions, seeds, and transpilation settings to re-run Aer and IBM_BRISBANE.
- **Statistics:** Time-series-aware DM and bootstrap tests → report *p*-values and confidence bands for model comparisons.
- **Noise & Hardware:** Aer noise models → **IBM_BRISBANE** runs with M3 mitigation.
- **Ablations:** Depth/entanglers/optimizer/shots.
- **Artifacts & Seeds:** All AAPL/MSFT predictions and test slices provided.

IX. CONCLUSION AND FUTURE WORK

On short horizons and modest sequence lengths L, compact QNN regressors built with Qiskit ML match a strong LSTM baseline while using fewer parameters and exhibiting statistically consistent generalization—provided the training loop is explicitly noise-aware and the final evaluation runs on well-calibrated IBM hardware. Our AAPL/MSFT experiments on simulator and on IBM_BRISBANE substantiate this claim and align with recent guidance in QML-for-finance on realistic noise modeling and shallow, data-reuploading PQCs [1]. The Qiskit-native stack (primitives, PyTorch integration) further reduces the sim-to-hardware gap, which we exploited for repeatable hardware validation. Taken together, these results demonstrate an engineering path—not just a theoretical promise - for deployable QML regressors in equity forecasting.

Future work:

(i) **Context-sharing QNNs:** Realize a multi-asset "share-and-specify" contextual ansatz that learns cross-asset structure (AAPL/MSFT/NVDA/GOOGL) on a single circuit, targeting parameter efficiency, better data efficiency.

(ii) **Quantum attention layers:** Add lightweight quantum-attention blocks to hybrid regressors and policy learners to stabilize training and improve risk-adjusted performance.
(iii) **Hybrid encoders:** Combine quantum Gramian/angular feature maps with shallow QNN heads to test whether quantum image-style embeddings improve short-horizon regression error.
(iv) **Federated QML:** Prototype privacy-preserving, cross-market training (exchanges/venues) via federated QNNs with noise-aware aggregation to respect data-sharing constraints while maintaining hardware realism.

REFERENCES

[1] M. Doosti, P. Wallden, C. B. Hamill, R. Hankache, O. T. Brown, and C. Heunen, "A Brief Review of Quantum Machine Learning for Financial Services," *arXiv preprint* arXiv:2407.12618, 2024. Available: https://arxiv.org/abs/2407.12618
[2] Z. Xu, Y. Wang, X. Feng, Y. Li, and H. Lin, "Quantum-Enhanced Forecasting: Leveraging Quantum Gramian Angular Field and CNNs for Stock Return Predictions," *arXiv preprint* arXiv:2310.07427, 2023. Available: https://arxiv.org/abs/2310.07427
[3] T. Fischer and C. Krauss, "Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions," *Expert Systems with Applications*, vol. 83, pp. 193–205, 2017 (printed 2018). DOI: 10.1016/j.eswa.2017.10.024
[4] K. Nakaji, S. Uno, Y. Suzuki, R. Raymond, T. Onodera, T. Tanaka, H. Tezuka, N. Mitsuda, and N. Yamamoto, "Approximate Amplitude Encoding in Shallow Parameterized Quantum Circuits and Its Application to Financial Market Indicator," *arXiv preprint* arXiv:2103.13211, 2021. Available: https://arxiv.org/abs/2103.13211
[5] S. Mourya, H. Leipold, and B. Adhikari, "Contextual Quantum Neural Networks for Stock Price Prediction," *arXiv preprint* arXiv:2503.01884, 2025. Available: https://arxiv.org/abs/2503.01884
[6] M. E. Şahin, E. Altamura, O. Wallis, S. P. Wood, A. Dekusar, D. A. Millar, T. Imamichi, A. Matsuo, and S. Mensa, "Qiskit Machine Learning: An Open-Source Library for Quantum Machine Learning," *arXiv preprint* arXiv:2505.17756, 2025. Available: https://arxiv.org/abs/2505.17756
[7] P. Pathak, V. Singh, S. Garg, and R. Kumar, "The Evolution of IBM's Quantum Information Software Kit (Qiskit): A Survey," *arXiv preprint* arXiv:2508.12245, 2025. Available: https://arxiv.org/abs/2508.12245
[8] N. Srivastava, S. Tiwari, and R. Singh, "The Potential of Quantum Techniques for Stock Price Prediction," *arXiv preprint* arXiv:2308.13642, 2023. Available: https://arxiv.org/abs/2308.13642
[9] L. Leclerc, A. Signoles, J. L. Guillot, et al., "Financial Risk Management on a Neutral Atom Quantum Processor," *arXiv preprint* arXiv:2212.03223, 2022. Available: https://arxiv.org/abs/2212.03223
[10] S. Dutta, V. R. Gowtham, and K. R. S. Krishnan, "QADQN: Quantum Attention Deep Q-Network for Financial Decision-Making," *arXiv preprint* arXiv:2408.03088, 2024. Available: https://arxiv.org/abs/2408.03088
[11] N. Meyer, J. Chen, and A. Shao, "Qiskit-Torch-Module: Fast Prototyping of Quantum Neural Networks," *arXiv preprint* arXiv:2404.06314, 2024. Available: https://arxiv.org/abs/2404.06314
[12] N. Innan, M. Al-Zafar Khan, and M. Bennai, "Quantum Federated Neural Network for Financial Fraud Detection (QFNN-FFD)," *Quantum Machine Intelligence*, vol. 6, no. 2, 2024. DOI: 10.1007/s42484-024-00118-9
[13] Y. Huang, S. X. Zhao, and L. Li, "Machine Learning for Stock Prediction Based on Fundamental Analysis," *arXiv preprint* arXiv:2202.05702, 2022. Available: https://arxiv.org/abs/2202.05702
[14] Y. Huang, S. Zhao, and L. Li, "Quantum-Machine Learning with Qiskit: Evaluating Regression Accuracy," *IET Quantum Communication*, vol. 4, no. 3, pp. 112–124, 2023. DOI: 10.1049/qtc2.12045
[15] IBM, "Quantum Portfolio Optimizer: A Qiskit Function," *IBM Quantum Documentation*, 2024. Available: https://quantum.ibm.com/docs/guides/finance/quantum-portfolio-optimizer
[16] F. X. Diebold and R. S. Mariano, "Comparing Predictive Accuracy," *J. Bus. Econ. Stat.*, vol. 13, no. 3, pp. 253–263, 1995. DOI: 10.1080/07350015.1995.10524599