A Quantum-Enhanced Transformer System for Real-Time Intraday Stock Forecasting

Abhinav Kochar, Sai Karthik Patri, Yagna Patel

Computer Science, School of Science and Engineering

UMKC School of Science and Engineering

Context and Motivation

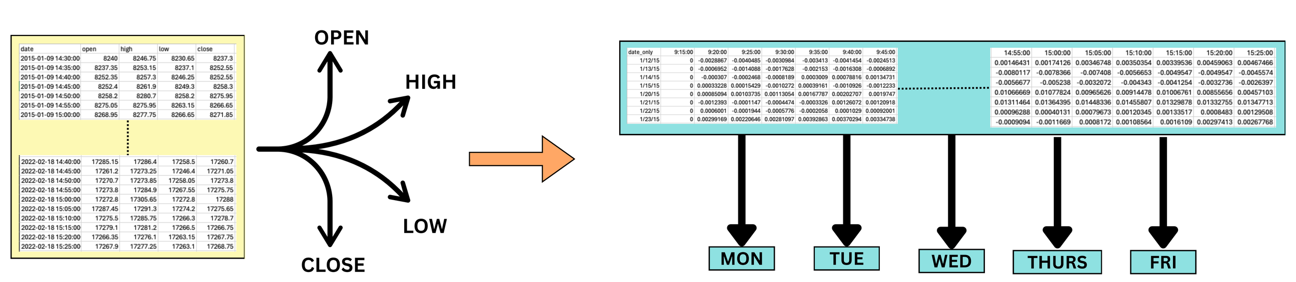
Intraday trading presents a fast-paced and volatile environment, where price fluctuations can be significant within just a few minutes. Accurately forecasting future prices in such settings is essential for algorithmic traders, market makers, and risk managers, as effective short-term predictions can lead to improved trade execution, risk mitigation, and profit optimization. However, existing forecasting models—such as **ARIMA**, classical **RNNs,** and even basic **LSTM** variants—often fall short in adapting to the rapid, non-linear, and time-sensitive nature of intraday data. These models typically struggle with capturing long-range dependencies, handling weekday-specific patterns, and adjusting to evolving market microstructures.

To address this challenge, we propose a forecasting framework that leverages the **Transformer architecture**, known for its ability to model sequential data using multi-head attention mechanisms. By incorporating day-wise and time-slot-specific segmentation strategies, the system is better equipped to recognize recurring intraday behaviors—such as the high volatility typical of Monday openings or the afternoon slowdowns on Fridays. Additionally, the integration of an optional quantum neural network extension introduces a novel dimension to the architecture, enabling the exploration of nonlinear and entangled relationships that may not be captured by classical models alone. Together, this approach aims to bridge the current gap in generating accurate, real-time, hour-ahead predictions within intraday financial markets.

Data Remodelling & Splitting

A key aspect of the methodology lies in comprehensively reorganizing raw **OHLC** data into structures that support high-resolution, feature-specific forecasting. Rather than treating all price information as one continuous time series, the pipeline splits each feature—Open, High, Low, and Close—into separate datasets. In these datasets, each row represents a full trading day, while columns correspond to uniform time intervals (e.g., **5-minute bars** from 9:15 AM to 3:25 PM). This transformation offers the model a clear view of intra-day patterns in individual price dimensions while maintaining the chronological order crucial for time-series modeling.

To further improve learning specificity and model generalization, we apply additional data restructuring and filtering techniques. For example, days are split by weekday and time block, enabling models to be trained on context-aware segments like "Monday Mornings" or "Wednesday Afternoons." These refinements reflect the inherent **non-stationarity** of financial markets, where behavior differs across sessions and weekdays. All datasets are normalized using either **log returns** or relative scaling based on the day's open price, helping reduce volatility bias and improve training convergence. The modular structure of this dataset setup also supports incremental real-time updates, making it ideal for live forecasting pipelines.



Day \ Slot	9:15–10:25 → 10:25–11:25	9:15–11:25 → 11:25–12:25	9:15–12:25 → 12:25–13:25	9:15–13:25 → 13:25–14:25	9:15–14:25 → 14:25–15:25
Monday	data_mon_slot1.csv	data_mon_slot2.csv	data_mon_slot3.csv	data_mon_slot4.csv	data_mon_slot5.csv
Tuesday	data_tue_slot1.csv	data_tue_slot2.csv	data_tue_slot3.csv	data_tue_slot4.csv	data_tue_slot5.csv
Wednesday	data_wed_slot1.csv	data_wed_slot2.csv	data_tue_slot3.csv	data_wed_slot4.csv	data_wed_slot5.csv
Thursday	data_thu_slot1.csv	data_thu_slot2.csv	data_thu_slot3.csv	data_thu_slot4.csv	data_thu_slot5.csv
Friday	data_fri_slot1.csv	data_fri_slot2.csv	data_fri_slot3.csv	data_fri_slot4.csv	data_fri_slot5.csv

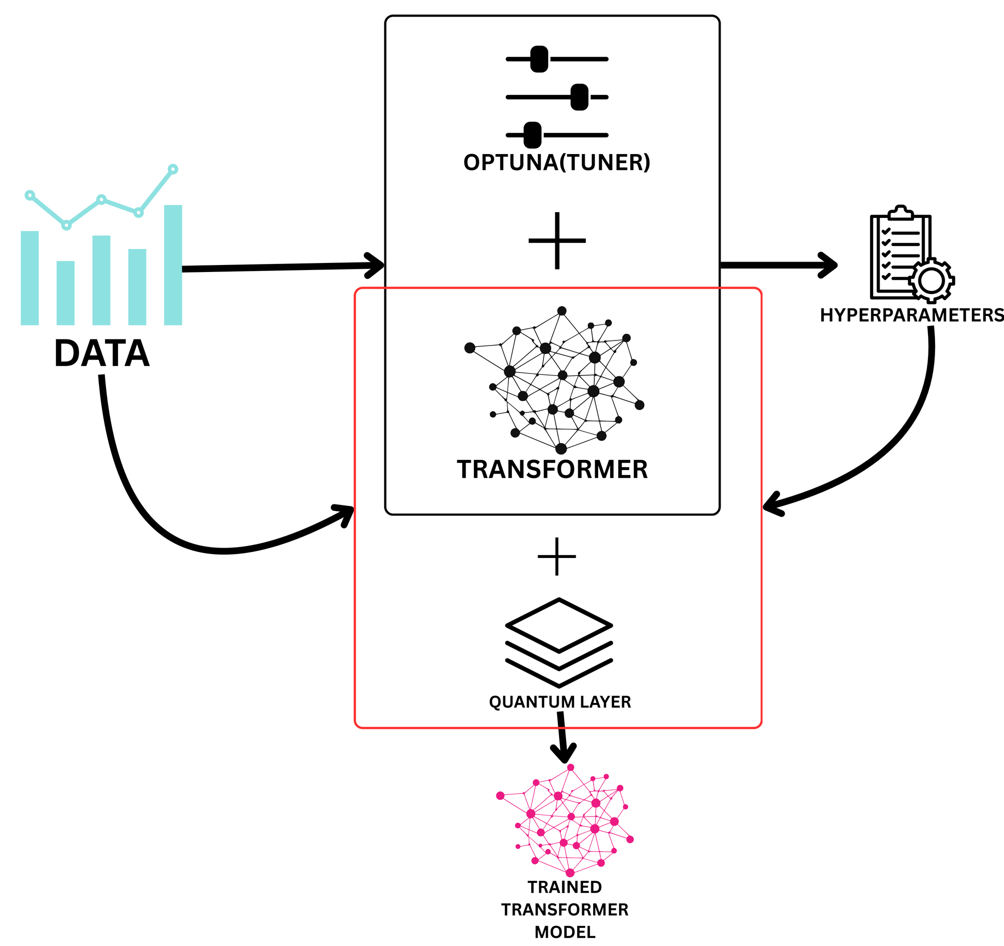
As shown in Figure 1, the raw stock data is separated into four distinct datasets by feature (O/H/L/C). Each dataset arranges days as rows and time slots as columns (9:15, 9:20, etc.). The small table within Figure 1 further illustrates the daily splits and file naming.

References

1. Vaswani, A., et al. (2017). "Attention Is All You Need." Advances in Neural Information Processing Systems.
2. Zhou, H., et al. (2021). "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting." arXiv preprint arXiv:2107.02737.
3. Schuld, M., et al. (2018). "Circuit-centric quantum classifiers." Physical Review A, 98(3), 032337.

Training Pipeline & Multiple Models

Once the data is neatly partitioned, the system proceeds to a training pipeline that begins with **hyperparameter tuning**, where parameters such as **learning rate, batch size,** and **attention heads** are systematically explored. This initial step ensures that each Transformer variant is matched with optimal configurations for its specific feature (Open, High, Low, or Close) and potential weekday/time splits. The Transformer core then takes center stage, parsing the time-sequenced data through multi-head attention layers that identify relevant signals across various segments of a given trading day. For users interested in pushing the boundaries of classical modeling, an optional **quantum neural network** layer can be appended, allowing partial transformations of the hidden representations into qubit states. Training involves iterative backpropagation, and for **quantum circuits**, a parameter-shift rule may be employed to tune the variational gates. The end result is a collection of specialized models—potentially one for each feature-day combination—each capturing the unique intraday patterns relevant to its assigned domain.



Day \ Slot	9:15–10:25 → 10:25–11:25	9:15–11:25 → 11:25–12:25	9:15–12:25 → 12:25–13:25	9:15–13:25 → 13:25–14:25	9:15–14:25 → 14:25–15:25
Monday	model_mon_slot1.h5	model_mon_slot2.h5	model_mon_slot3.h5	model_mon_slot4.h5	model_mon_slot5.h5
Tuesday	model_tue_slot1.h5	model_tue_slot2.h5	model_tue_slot3.h5	model_tue_slot4.h5	model_tue_slot5.h5
Wednesday	model_wed_slot1.h5	model_wed_slot2.h5	model_wed_slot3.h5	model_wed_slot4.h5	model_wed_slot5.h5
Thursday	model_thu_slot1.h5	model_thu_slot2.h5	model_thu_slot3.h5	model_thu_slot4.h5	model_thu_slot5.h5
Friday	model_fri_slot1.h5	model_fri_slot2.h5	model_fri_slot3.h5	model_fri_slot4.h5	model_fri_slot5.h5

Figure 2 outlines our training pipeline. First, the tuner identifies optimal hyperparameters for the Transformer. Then, the data and chosen hyperparameters feed into the Transformer (optionally combined with a quantum neural network). After training completes, we obtain a specialized model for each feature (Open, High, Low, Close) and, if applicable, each day/time split.

Results and Quantum Enhancements

To assess the system's performance, visual and quantitative evaluations focus on both training outcomes and inference quality. Heatmaps of **mean absolute error (MAE)** or other relevant metrics provide a panoramic view of how errors vary across different time slots, weekdays, and individual features (Open, High, Low, Close). These **heatmaps** often reveal distinct patterns—such as higher error spikes during morning hours or post-lunch volatility—offering hints for further model refinement. Additionally, **loss curves** track the evolution of training for select configurations, illustrating how quickly the Transformer converges and whether the optional quantum component yields lower final losses or faster convergence. Sample **inference** plots, overlaying predicted vs. actual prices, demonstrate how well the system captures intraday fluctuations under both historical backtesting and live conditions. In certain cases, the **quantum-augmented approach** can uncover subtle correlations leading to incremental gains in accuracy, although these improvements may be context-dependent. Collectively, these analyses help highlight the robustness and adaptability of the hour-ahead forecasting method.

In Figure 3, we compare the mean absolute error across different models and time slots. Darker (or lighter) squares indicate higher (or lower) forecasting error, revealing which hours or weekdays are more challenging.

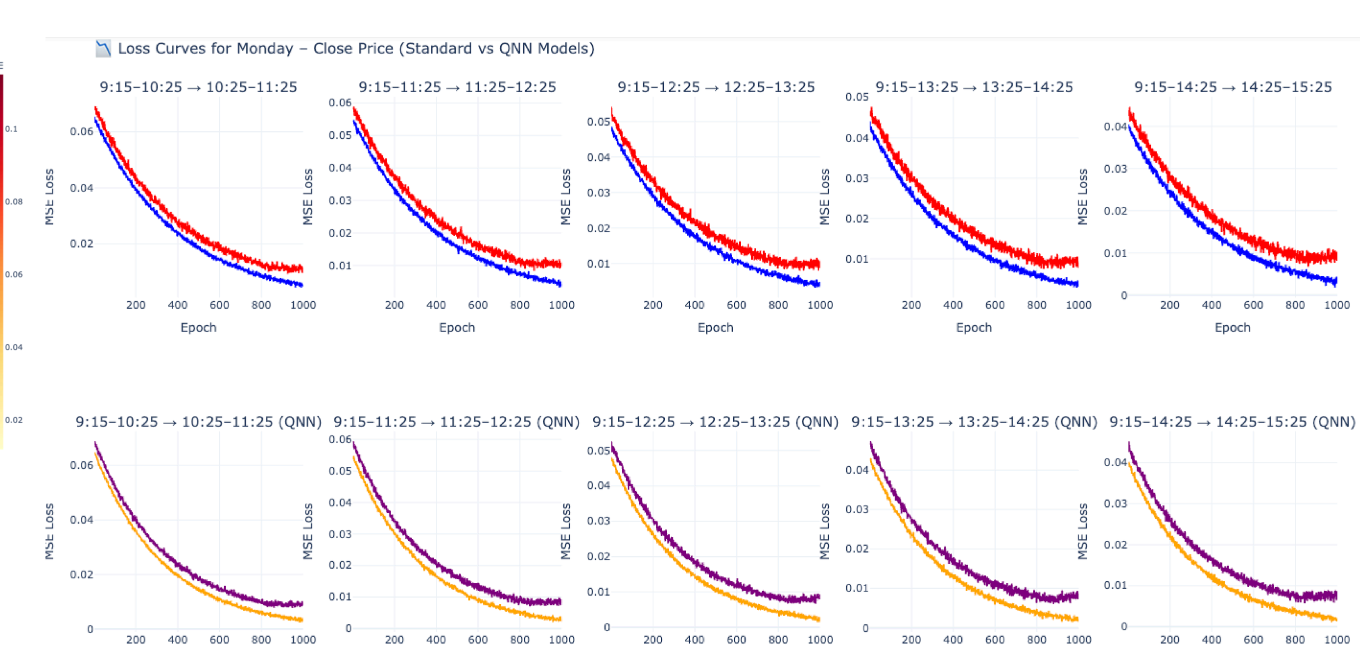
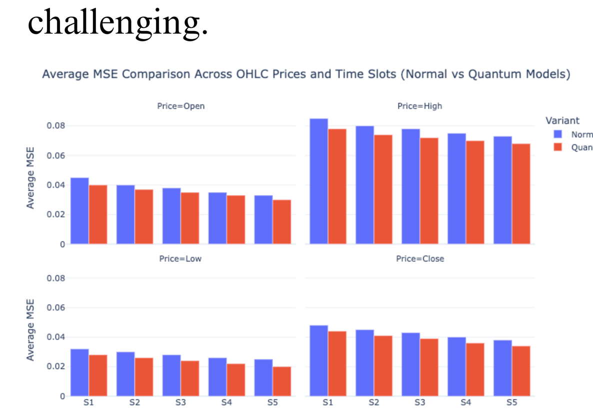


Figure 4 depicts our training loss progression for Monday's Close price. The top row shows standard Transformer curves for five distinct time blocks, while the bottom row shows the Transformer + QNN variant. In most cases, the quantum layer converges faster or to a lower loss.



Quantum vs. Classical Model Performance Across Time Slots

In Figure 5, Comparison of average MSE across Open, High, Low, and Close prices for five intraday time slots (S1–S5), using both standard Transformer (blue) and Transformer + Quantum models (red). The quantum-enhanced models consistently show slightly lower error across all price types, with the most noticeable improvements in predicting High prices, indicating better handling of intraday volatility.

Architecture and Inferencing

In a live trading or real-time analytic context, the trained models are orchestrated by an **inference engine** designed to handle hourly predictions with minimal lag. As new market data arrives, the system instantly directs each OHLC component to the appropriate **normalization** routine—usually the same log-return or open-based scaling applied during training—before storing these values in an incremental data file. From there, a **Model Selector** module identifies which specialized Transformer (potentially augmented by the quantum layer) should be deployed based on the current day and time segment. The selected model then processes the freshly normalized data, generating a raw forecast for the next hour's price movement. This output is **denormalized** to restore realistic price levels and subsequently aggregated into a **candlestick representation**, combining predicted Open, High, Low, and Close. In parallel, the model's forecasts are logged in an incremental output file, creating a continuous, time-stamped record of both predicted and actual market values. This rolling window design thus seamlessly integrates short-term forecasting into ongoing market operations.

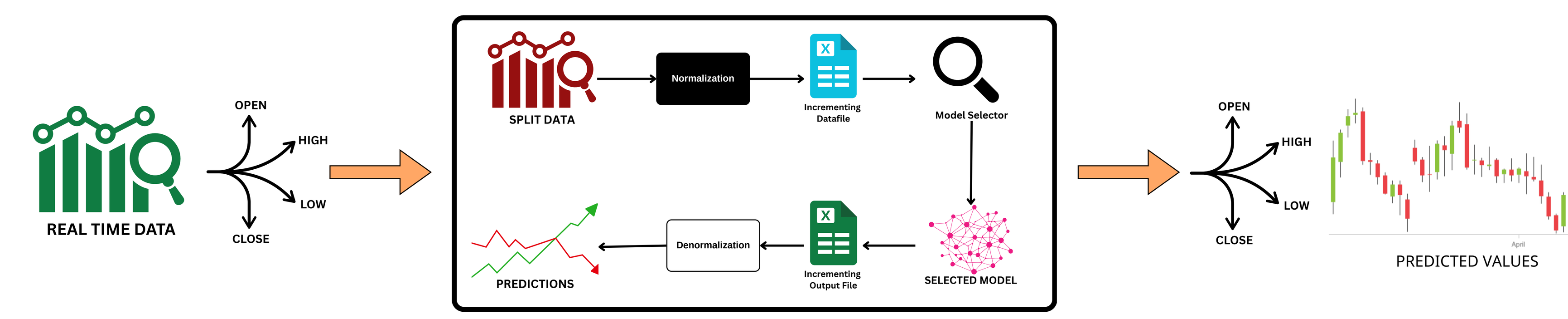
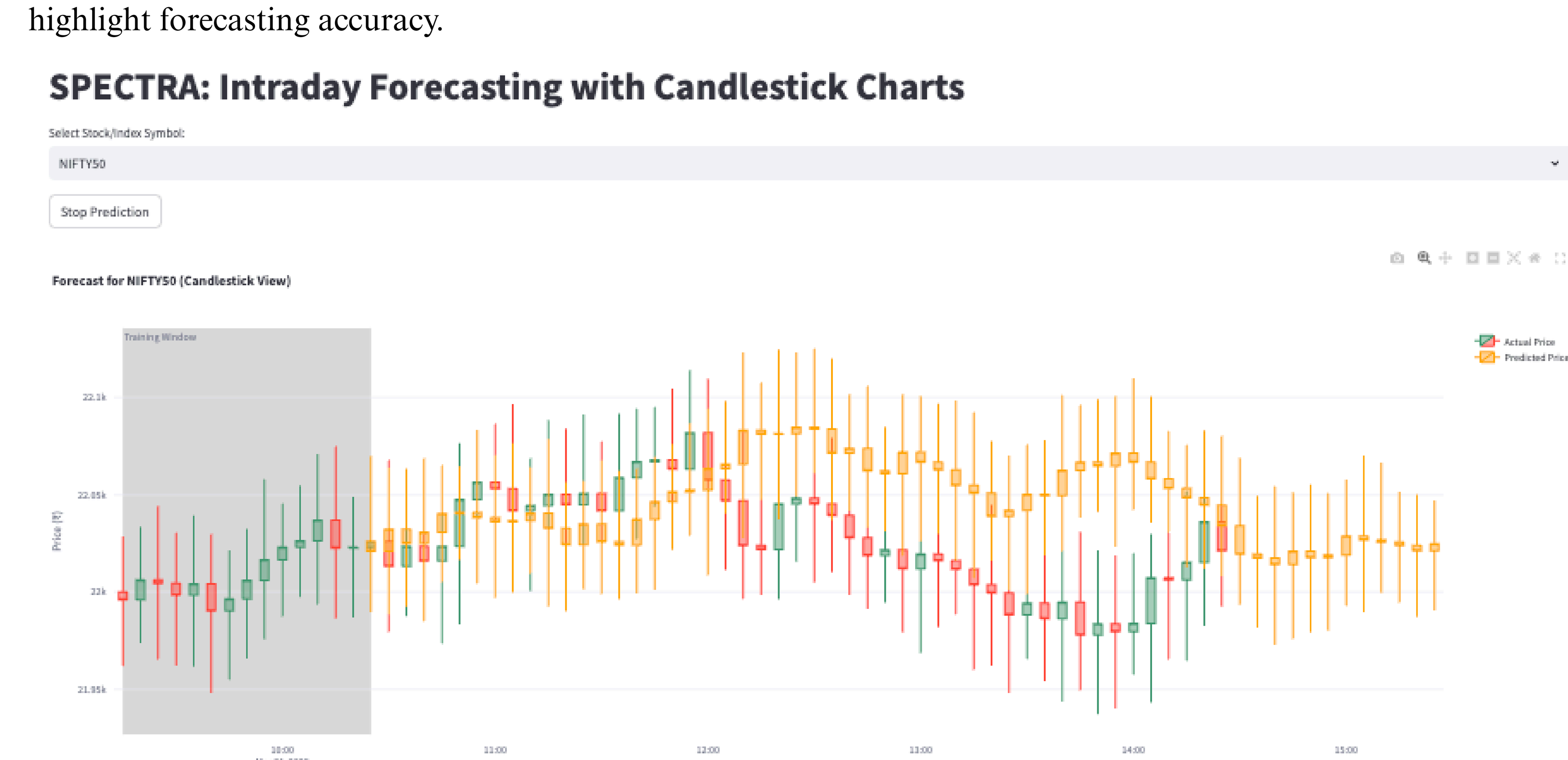


Figure 6 highlights the system's live architecture. As intraday data arrives, it's split into separate features, normalized, and appended to an incrementing data file. The Model Selector chooses which pretrained model to deploy, generating predictions that are subsequently denormalized and converted into candlestick format

Real Time Application Interface

To evaluate the system's forecasting performance, both quantitative metrics and visual outputs were analyzed. The **Quantum-augmented Transformer** models consistently achieved lower mean squared errors (MSE) across all OHLC price types and intraday time slots when compared to standard Transformers, as shown in Figure 5. This improvement was particularly significant for High price predictions, which are typically more volatile and harder to model.

For real-time deployment, a fully functional application interface was developed, powered by live data from the **Yahoo Finance API** and using the Nifty 50 index as the primary data source. The interface dynamically displays both predicted and actual price movements in real time, using interactive line plots and **candlestick charts** to highlight forecasting accuracy.



In Figure 7, The forecasting interface visualizes actual and predicted intraday stock prices using candlestick charts, with the initial training window (9:15–10:25 AM) shaded in grey. Predictions are made one hour ahead of the current time, allowing the model to **anticipate future price trends**, and are displayed from 10:25 AM to market close for real-time comparison with actual movements.

Acknowledgements

The authors gratefully acknowledge the contributions of Dr. Yugyung Lee, whose expertise in Transformer architectures and quantum computing provided critical guidance throughout the model development process. We also extend our thanks to [institution/lab name], whose computational resources made extensive hyperparameter tuning feasible. The support and collaboration of Udiptaman Das, Luke Miller, and Yagna Patel proved invaluable during data cleansing and early-stage analyses. Lastly, the project benefited greatly from open-source communities (e.g., [framework/tool library] teams), whose tools and repositories accelerated our research and experimentation.