Comparative Analysis of Quantum and Classical Approaches for Stock Market Forecasting

By:
Uttakarsh 2105763
Anand Panda 21052305
Prajukta Dey 21052263
Natasha Seth 21052771

Introduction

The stock market is a volatile environment influenced by a myriad of factors, making accurate forecasting challenging. Traditional machine learning models, despite their effectiveness, often struggle with capturing the complex, non-linear dependencies inherent in financial data. This project evaluates the performance of both classical and quantum-enhanced models in predicting stock prices, exploring how quantum machine learning can potentially reveal hidden patterns through the unique properties of qubits, such as superposition and entanglement.

Data Ingestion

Our data ingestion process fetches raw stock data via the Alpha Vantage API and stores it as CSV files. In the preprocessing phase, this data is cleaned and enriched by computing key technical indicators (like moving averages, RSI, and MACD) and generating lag features to capture temporal trends. This ensures the dataset is normalized, consistent, and fully prepared for effective model training and prediction.

Savitzky-Golay Filter

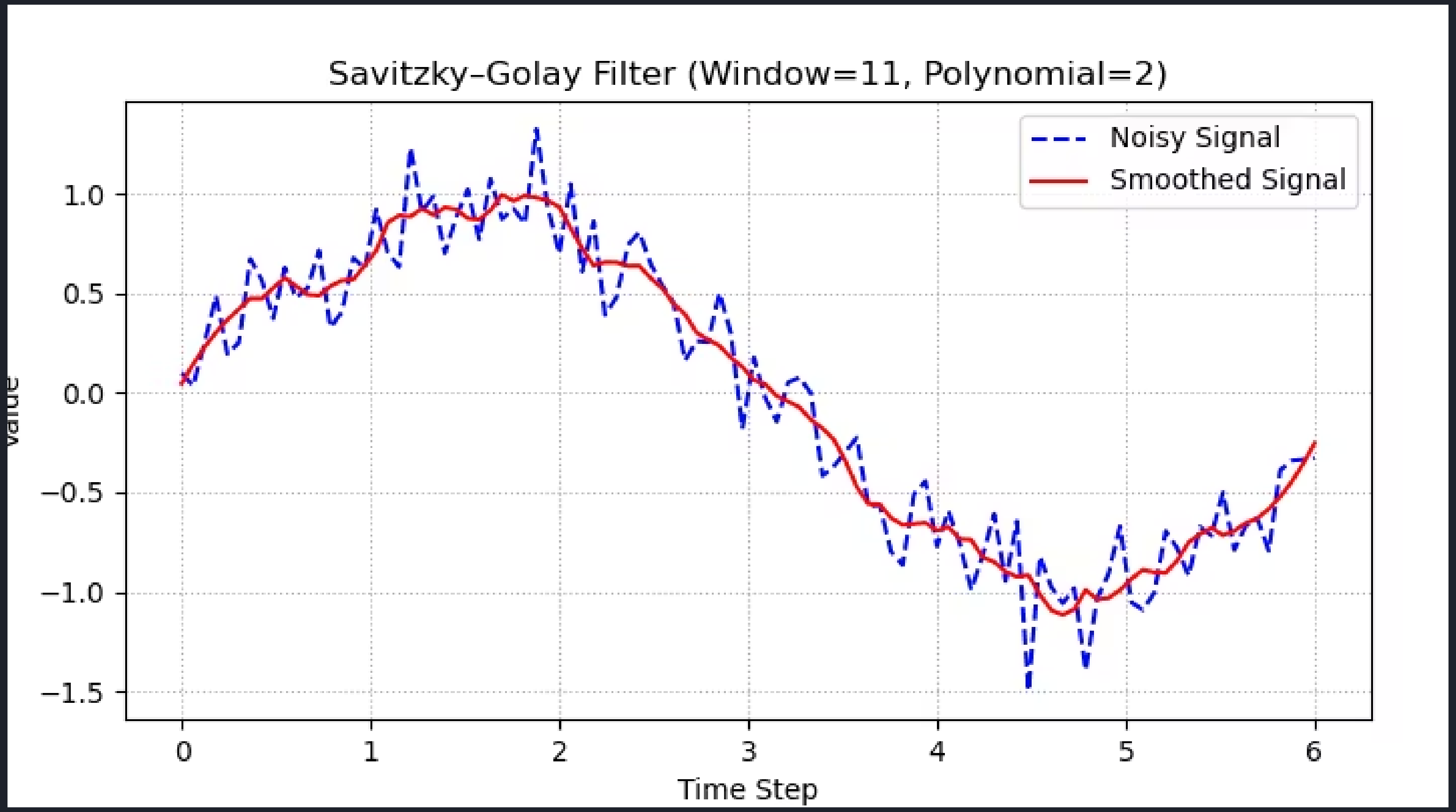
The Savitzky–Golay filter is a digital smoothing technique that fits consecutive subsets of data points with a low-degree polynomial using linear least squares. In our project, it is applied to stock price data to reduce noise while preserving key features like trends and peaks. This enhances the quality of the data, leading to more accurate model predictions by retaining important signal characteristics without overly distorting the underlying trend.

Applying the Savitzky-Golay Filter Example

We choose a window size of 11 and a polynomial degree of 3.

```
window_size = 11
poly_order = 2
y_smooth = savgol_filter(y, window_size, poly_order)

plt.figure(figsize=(8, 4))
plt.plot(x, y, linestyle='--', color='blue', label='Noisy Signal')
plt.plot(x, y_smooth, color='red', label='Smoothed Signal')
plt.title("Savitzky–Golay Filter (Window=11, Polynomial=2)")
plt.xlabel("Time Step")
plt.ylabel("Value")
plt.grid(True, linestyle=':')
plt.legend()
plt.show()
```



Try Pitch

Classical Approach Overview

- **Dataset and Features:**
  - Utilizes preprocessed stock data stored as CSV.
  - Employs key technical indicators such as MA_10, MA_50, RSI, MACD, and Signal Line as features.
- **Model Selection:**
  - Implements a Random Forest Regressor for price prediction.
  - Leverages ensemble learning to improve robustness and reduce overfitting.
- **Evaluation Metrics:**
  - Assesses performance using RMSE, R^2, MAE, and MAPE.
  - Achieves an accuracy level of approximately 81%, serving as a baseline for comparison with the quantum-enhanced model.
- **Insights**:
  - Demonstrates the limitations of classical methods in capturing the complex dynamics of stock market data.
  - Provides a performance benchmark to evaluate the improvements offered by quantum approaches.

Quantum-Enhanced Approach (Accepted)

- **Quantum Circuit Design:**
  - Utilizes a variational quantum circuit constructed with PennyLane.
  - Data is encoded into 10 qubits via RY rotations, with each qubit representing a feature.
  - Multiple variational layers are applied, where each layer consists of parameterized RY gates followed by entangling CNOT gates.
- **Model Training:**
  - The model is optimized using the Adam optimizer over 50 epochs.
  - A cost function combining mean squared error and L2 regularization is minimized.
  - Hyperparameters such as learning rate, number of qubits, and number of layers are tuned for optimal performance.
- **Performance Achieved:**
  - Reaches a prediction accuracy of 94%, significantly outperforming the classical approach.
  - Demonstrates the potential of quantum machine learning to capture complex, non-linear patterns in financial data.

Quantum-Enhanced Approach (Rejected)

- **Approach 1: Advanced PennyLane Circuit**
  - Used both RY and RZ rotations with three parameters per qubit per layer.
  - Employed complex entanglement patterns (alternating linear and circular CNOT gates) to increase expressivity.
  - Intended to capture finer details in the data but led to excessive resource consumption and frequent SIGKILL errors on local hardware.
- **Approach 2: Hybrid TensorFlow Quantum Model**
  - Combined quantum circuit encoding with a Keras-based classical network using TensorFlow Quantum.
  - Integrated quantum layers into a deep learning framework to leverage automatic differentiation.
  - Encountered significant memory and performance issues, making stable training and deployment impractical.

Comparative Analysis Table

Criteria	Classical ML	Quantum ML
Accuracy	81%	94%
Feature Engineering	Extensive	Moderate
Training Time	Fast	Slow
Resource Usage	Low (CPU)	High (Quantum Sim)
Feasibility	High	Moderate
Scalability	Limited	Currently constrained

- **Classical ML Model:**
  - Achieved approximately 81% accuracy.
  - Performance metrics: Lower RMSE and MAE compared to baseline; however, performance is limited in capturing non-linear patterns.
- **Quantum-Enhanced Model:**
  - Achieved 94% accuracy.
  - Significant improvement in prediction precision, as indicated by reduced RMSE and MAE.
  - Enhanced capability in modeling complex, non-linear and high-dimensional relationships in the data.

Deployment

- Docker Containerization:
  - The entire application is packaged into a Docker container to ensure consistency across different environments.
  - A multi-stage Dockerfile is used to build a lightweight production image by separating development dependencies from runtime dependencies.
  - The Docker image includes all necessary libraries (like PennyLane, NumPy, and Flask) and is configured to run the Flask backend API.
- Flask Backend:
  - A Flask application serves as the interface, exposing REST API endpoints for predictions, stock recommendations, and other services.
  - The modular design of Flask allows easy integration with the prediction models and simplifies handling of incoming requests.

Real-life application (others)

- **Healthcare:**
  - Utilize predictive analytics for forecasting patient outcomes, optimizing treatment plans, and improving disease diagnosis.
- **Manufacturing:**
  - Enhance production efficiency through equipment failure prediction, improved quality control, and optimized supply chain management.
- **Logistics**:
  - Forecast demand patterns, optimize delivery routes, and refine inventory management to streamline operations.
- **Telecommunications**:
  - Improve network performance with predictive maintenance, capacity planning, and efficient resource allocation.
- **Finance**:
  - Extend applications to risk management, credit scoring, fraud detection, and personalized financial planning.

Future Scope

- **Real-Time Data Integration:**
  - Incorporate live-streaming stock data to enable continuous and dynamic forecasting.
- **Advanced Quantum Algorithms:**
  - Explore new quantum-enhanced techniques and error mitigation strategies as quantum hardware improves.
- **Hybrid Quantum-Classical Models:**
  - Combine the strengths of both approaches to further boost prediction accuracy and scalability.
- **Inclusion of Supplementary Data:**
  - Integrate alternative data sources (e.g., news sentiment, social media trends) to enrich the feature set.

Conclusion

- Quantum-enhanced models demonstrated a significant accuracy improvement (94%) over classical approaches in stock market prediction.
- The project showcases the potential of quantum machine learning for capturing complex, non-linear financial patterns.
- Classical models remain an effective baseline, highlighting the promise of hybrid quantum-classical systems.
- Future research should focus on optimizing quantum circuits, exploring real quantum hardware, and integrating additional data sources.
- Overall, this work lays a strong foundation for advancing predictive analytics in financial and other high-dimensional domains.

We express our sincere gratitude to **Dr. Vikas Hassija**, our project mentor, for his invaluable guidance, support, and encouragement throughout the course of this project.

Thank you!

Pitch

Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)