High-Frequency Algorithmic Trading

With Deep Learning

Ritesh Kumar

2024SP_MS_DSP_498-DL_SEC61: Capstone

Module 2

Project and Testing Plan including Data Schema

Kimberly Chulis and Srabashi Basu

June 27, 2024

Table of Contents

Abstract

In the dynamic landscape of financial markets, deep learning techniques have revolutionized algorithmic trading strategies. This project aims to develop an advanced algorithmic trading system for BankNifty, a prominent Indian banking index. The system will leverage historical data, including Cumulative Open Interest (COI), Price, Volume, India Volatility Index, and Technical Indicators such as Moving Averages, RSI, and MACD, to predict market trends and make informed trading decisions.

The primary objective is to design a sophisticated trading system that analyzes historical data and informs trading decisions using deep learning models. Key components of the project include Feature Engineering, Backtesting, Risk Management, Performance Metrics, Scalability, and Adaptation. High-frequency data collection at 1-minute intervals will enable the system to capture short-term market movements and trends.

Data preprocessing steps will include handling missing data, feature engineering to create new features, data normalization, and splitting the dataset into training, validation, and testing sets. The deep learning models employed will include Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNN), and Transformer Models, each chosen for their ability to capture sequential patterns, identify complex features, and handle long-range dependencies in time-series data.

Effective feature engineering will be crucial, with techniques like recursive feature elimination (RFE) and feature importance ranking identifying the most relevant features. Technical indicators will enhance the feature set. Rigorous testing will evaluate the trading strategy's performance on historical data, and optimization will fine-tune parameters to maximize profitability and minimize risk.

Risk management strategies, including stop-loss orders and position sizing, will protect capital and minimize losses. Performance metrics such as the Win-Loss Ratio and Profit Factor will provide a comprehensive assessment of the system's effectiveness.

Designed for scalability and adaptability, the system can be applied to other indices and financial instruments. Periodic retraining and continuous monitoring will maintain model accuracy and relevance, ensuring the system remains effective and responsive to market dynamics. This project aims to create a powerful tool for navigating the complexities of BankNifty trading, with principles extendable to other financial markets.

Project Plan

The project plan is structured to ensure a systematic approach to development and deployment. It begins with the publication of the project charter, clearly defining the scope, objectives, and deliverables. Setting up a live-data extraction system is crucial to gather the required historical data, which includes Cumulative Open Interest (COI), Price, Volume, and various Technical Indicators.

The next phase involves conducting exploratory data analysis to understand data distributions, identify trends, and uncover any quality issues that need addressing. Preprocessing the data for modeling is a critical step, involving feature engineering and data normalization to prepare the dataset for effective model training.

Exploration of different deep learning techniques, such as LSTM, CNN, and Transformer models, forms the core of the project. Evaluating these models and comparing their performance will help in identifying the most suitable approach for our trading system. Fine-tuning hyperparameters and validating the models on a separate validation dataset ensures robustness and reliability.

Once the best performing model is selected, the deployment phase will involve integrating the model into a live trading environment. Continuous monitoring and adjustment based on real-time data feedback will be essential to maintain optimal performance.

The final stages of the project involve compiling the progress and results, presenting the outcomes. This structured plan aims to deliver a powerful tool for high-frequency trading, tailored to the complexities of BankNifty and potentially extendable to other indices.

Weeks	**Deliverables**	**Completion Dates**	**Status**
1.0	Publish Project Charter	24 June 2024	Completed
1.1	*Define Project Scope and Objectives*		
1.2	*Outline Project Deliverables*		
2.0	Set-up Live-Data Extraction System	01 July 2024	In-Progress
3.0	Conduct the Exploratory Data Analysis	08 July 2024	Planned
3.1	*Perform Initial Data Analysis*		
3.2	*Visulaize Data Distributions and Trends*		
3.3	*Identify any Data Quality Issues*		
4.0	Pre-process Data for Moedlling	15 July 2024	Planned
4.1	*Engineer New Features*		
4.2	*Normalize Data*		
5.0	Explore Modelling Techniques	22 July 2024	Planned
5.1	*Evaluate different deep learning models (LSTM, CNN, Transformers)*		
5.2	*Perform model comparison*		
5.3	*Document findings and insights*		
6.0	Identify the Best Model	29 July 2024	Planned
6.1	*Fine-tune hyperparameters of selected models*		
6.2	*Validate models on validation dataset*		
6.3	*Select the best performing model*		
7.0	Deploy the Best Model	05 August 2024	Planned
7.1	*Deploy model in live trading environment*		
7.2	*Monitor model performance*		
7.3	*Make necessary adjustments based on live data feedback*		
8.0	Report Progress	12 August 2024	Planned
8.1	*Compile project progress and results*		
8.2	*Present Progress*		

1. Data Collection and Preprocessing:

- Data Collection:
    - Source: Reliable financial data providers.
    - Frequency: Every 1 minutes.
    - Data Points: Cumulative Open Interest (put and call), Price (open, high, low, close), Volume (put and call), India Volatility Index.
- Data Preprocessing:
    - Missing Data Handling: Imputation or removal.
    - Feature Engineering: Calculation of moving averages, exponential moving averages, RSI, MACD, and other technical indicators.
    - Data Normalization: Min-max scaling or Z-score normalization.
    - Train-Validation-Test Split:
        - Training: 70%
        - Validation: 20%
        - Testing: 10%

2. Deep Learning Models:

- LSTM Networks: To capture sequential patterns and dependencies in time-series data.
- CNNs: To identify patterns in time-series data by treating them as temporal sequences.
- Transformer Models: To handle long-range dependencies and capture intricate patterns.

3. Feature Engineering and Selection:

- Techniques: Recursive Feature Elimination (RFE), feature importance ranking.
- Technical Indicators: Moving Averages, RSI, MACD.

4. Optimization:

- Fine-tuning entry and exit criteria, stop-loss thresholds, position sizing.

5. Risk Management:

- Strategies: Stop-loss orders, position sizing, portfolio diversification.

6. Performance Metrics:

- Win Loss Ratio $\left(\frac{Number\ of\ Winning\ Trades}{Number\ of\ Losing\ Trades}\right)$
- Profit Factor $\left(\frac{Total\ Profit\ from\ Winning\ Trades}{Total\ Loss\ from\ Losing\ Trades}\right)$

7. Scalability and Adaptation:

- Design: Scalable and adaptable for other indices and financial instruments.
- Retraining: Periodic retraining on updated data.
- Monitoring: Continuous monitoring and adjustment based on market conditions.

8. Conclusion:

- Focus: Develop a powerful tool for BankNifty trading.
- Versatility: Extendable to other indices and financial instruments.

Testing Plan

A robust testing plan is essential to ensure that the algorithmic trading system for BankNifty meets detailed requirements and performs reliably. This testing plan encompasses various testing types, including Unit Testing, System Testing, Integration Testing, and User Acceptance Testing (UAT), each targeting specific aspects of the system to validate its functionality, performance, and user satisfaction.

1. Testing Overview: The primary purpose of this testing phase is to ensure the system meets the specified requirements and performs reliably in real-world scenarios. The types of testing involved include Unit Testing, System Testing, Integration Testing, and User Acceptance Testing (UAT).
2. Unit Testing: The objective of unit testing is to verify the functionality of individual components and functions. The key scenarios and test cases include:
    - Data Collection: Verifying the correct retrieval of data from a financial data provider.

- Data Preprocessing: Validating the handling of missing data, feature engineering processes, and data normalization techniques.
- Model Training: Ensuring the correct implementation of LSTM, CNN, and Transformer models.
- Prediction: Checking the accuracy of short-term price predictions.

3. System Testing: System testing aims to ensure that the entire system works cohesively as intended. The scenarios and test cases for this phase include:
   - End-to-End Workflow: Testing the complete process from data collection to prediction.
   - Performance Metrics Calculation: Validating the computation of key performance metrics such as the Sharpe Ratio and Maximum Drawdown.
   - Backtesting Framework: Ensuring accurate simulation of the trading strategy against historical data.
4. Integration Testing: The objective of integration testing is to verify the seamless integration of different system modules. Key scenarios and test cases include:
   - Data Pipeline: Testing the integration between data collection, preprocessing, and model training modules.
   - Model Integration: Ensuring the smooth integration of LSTM, CNN, and Transformer models with the overall trading strategy.
   - Risk Management Integration: Validating the implementation of risk management strategies such as stop-loss orders and position sizing.
5. User Acceptance Testing (UAT): The goal of UAT is to ensure that the system meets user expectations and requirements. Scenarios and test cases include:
   - User Scenarios: Testing the system on live-market data, without initiating any real trades.

- Adjustments: Making necessary adjustments based on user feedback to improve the system.

6. Test Reports: Test reports will contain detailed information, including sample/test data, inputs, expected outputs, and actual results. These reports will be provided as part of the regular status updates.

This comprehensive testing plan ensures that every aspect of the algorithmic trading system is thoroughly validated, guaranteeing its reliability and performance in real-world trading environments.

Data Schema

Data Extraction Process: The dataset is compiled from two primary data streams to ensure comprehensive coverage of market activity:

1. First Data Stream: This stream includes key price data and volatility index (VIX). It generates data with a frequency range of 10 to 120 times per minute. The data points include:
   - open
   - high
   - low
   - close
   - vix

   The high-frequency data from this stream is summarized every minute to produce a consolidated view that captures the essential price movements and volatility within that minute.

2. Second Data Stream: This stream focuses on complete options chain data, generating data once every two to three minutes. The data points include:
    - total_vol
    - ce_vol (Call option volume)
    - pe_vol (Put option volume)
    - total_oi (Total open interest)
    - ce_oi (Call option open interest)
    - pe_oi (Put option open interest)
    - oi_chg (Open interest change for both call and put options)

This summarized data from the first stream is matched with the most recent summarized options chain data from the second stream to ensure synchronization.

Technical Indicators Calculation: After the synchronization of the two data streams, various technical indicators are computed and added to the dataset. These indicators, derived from both price and volume data, provide deeper insights into market trends, momentum, and potential reversal points. Examples include moving averages (SMA and EMA), MACD, RSI, Bollinger Bands, ATR, and others as listed in the schema.

This comprehensive approach to data extraction and feature engineering ensures that the trading model has access to high-quality, relevant data, enabling accurate predictions and effective trading decisions.

The data schema includes a comprehensive set of features designed to capture various aspects of market activity and technical indicators. Below is a detailed breakdown of each feature:

- date: The specific date of the trading data.
- time: The specific time of the trading data.
- expiry_day: The expiration date for options or futures contracts.

- open: The opening price of the asset.
- high: The highest price of the asset during the trading period.
- low: The lowest price of the asset during the trading period.
- close: The closing price of the asset.
- total_vol: Total trading volume.
- ce_vol: Call option volume.
- pe_vol: Put option volume.
- vol_chg: Change in trading volume.
- total_oi: Total open interest.
- ce_oi: Call option open interest.
- pe_oi: Put option open interest.
- oi_chg: Change in open interest.
- vix: Volatility Index.
- SMA_5: 5-period Simple Moving Average.
- SMA_10: 10-period Simple Moving Average.
- SMA_20: 20-period Simple Moving Average.
- EMA_12: 12-period Exponential Moving Average.
- EMA_20: 20-period Exponential Moving Average.
- RSI: Relative Strength Index.
- EMA_26: 26-period Exponential Moving Average.
- MACD: Moving Average Convergence Divergence.
- Signal_Line: Signal line for MACD.
- MACD_Histogram: Histogram for MACD.
- Middle_Band: Middle band of Bollinger Bands.
- Upper_Band: Upper band of Bollinger Bands.

- Lower_Band: Lower band of Bollinger Bands.
- %K: Stochastic Oscillator %K.
- %D: Stochastic Oscillator %D.
- ATR: Average True Range.
- OBV: On-Balance Volume.
- VWAP: Volume-Weighted Average Price.
- ROC_10: Rate of Change over 10 periods.
- ROC_20: Rate of Change over 20 periods.
- CCI: Commodity Channel Index.
- TR: True Range.
- DX: Directional Movement Index.
- ADX: Average Directional Index.
- Pivot: Pivot point for support and resistance levels.
- R1: First resistance level.
- S1: First support level.
- R2: Second resistance level.
- S2: Second support level.
- R3: Third resistance level.
- S3: Third support level.
- nxt_opn: Next period opening price.
- nxt_high: Next period highest price.
- nxt_low: Next period lowest price.
- nxt_close: Next period closing price.

The selected factors in this data schema are comprehensive and tailored to capture the multifaceted nature of financial markets, making them highly suitable for developing a robust

algorithmic trading model. Firstly, the inclusion of basic price data (open, high, low, close) and trading volumes (total_vol, ce_vol, pe_vol) provides the fundamental information necessary to understand market behavior and trading activity. These raw data points are essential for any financial model as they reflect the primary movements and liquidity in the market.

Moreover, the dataset includes a range of technical indicators, such as Simple Moving Averages (SMA), Exponential Moving Averages (EMA), and the Relative Strength Index (RSI). These indicators are crucial for identifying trends, momentum, and potential reversals in the market. For instance, moving averages smooth out price data to help identify the direction of the trend, while RSI provides insights into overbought or oversold conditions. Including various periods for these indicators (e.g., SMA_5, SMA_10, SMA_20, EMA_12, EMA_20) allows the model to capture both short-term and long-term market trends.

Advanced technical indicators like MACD, Bollinger Bands, and Average True Range (ATR) add further depth to the model. MACD helps in understanding the momentum and potential reversal points, while Bollinger Bands provide a visual representation of volatility and potential breakout points. ATR is crucial for assessing market volatility, which is essential for risk management and setting appropriate stop-loss levels.

The dataset also includes volatility indices (vix) and derivative data (ce_oi, pe_oi), which are vital for understanding market sentiment and the underlying risk in the market. Factors such as Directional Movement Index (DX), Average Directional Index (ADX), and Commodity Channel Index (CCI) help in identifying the strength and direction of market trends, adding another layer of analysis for the model.

Lastly, the inclusion of next period prices (nxt_opn, nxt_high, nxt_low, nxt_close) facilitates the training and evaluation of predictive models, allowing for direct assessment of the

model's forecasting ability. By incorporating these diverse factors, the model can leverage a holistic view of the market, enhancing its ability to predict price movements and execute trades effectively. This comprehensive feature set ensures that the model is equipped to handle various market conditions and make informed trading decisions.

References

1. Manveer Kaur Mangat, Erhard Reschenhofer, Thomas Stark, Christian Zwatz. High-Frequency Trading with Machine Learning Algorithms and Limit Order Book Data. Data Science in Finance and Economics, 2022, 2(4): 437-463. (Link)
2. Arévalo, Andrés & Nino, Jaime & Hernandez, German & Sandoval, Javier. (2016). High-Frequency Trading Strategy Based on Deep Neural Networks. 9773. 424-436. 10.1007/978-3-319-42297-8_40. (Link)