Machine Learning for Trading and Portfolio Optimization using LSTM and Quantitative Methods

Rak Kantarak
Individual Interested Research

1. Introduction

This is my own project that I made because I am interested in AI and finance. I wanted to test how machine learning (LSTM) can be used together with portfolio theory to help in investment decisions.

The project is personal research, not for work. I used this project to learn more about data, trading, and quantitative finance.

2. Method

Steps I used in this project:

1. Data – I collected price data from gold, silver, oil, wheat, and Thai stocks.
2. Features – I used price and RSI (Relative Strength Index).
3. Model – I trained an LSTM model with PyTorch.
4. Optimization – I used Optuna to find the best hyperparameters.
5. Signals – Predictions are changed into buy/sell/hold signals.
6. Backtest – I tested with transaction cost and next-bar execution.
7. Portfolio – I compared Tangent Portfolio (MPT) with Dynamic Portfolio (HJB).
8. Metrics – I measured return %, drawdown %, volatility, and win rate.

3. Results (Work in Progress) This project is still ongoing research.

Early experiments show that:

- The LSTM + RSI model can generate trading signals, with some periods showing profit even after transaction costs.
- The Dynamic Portfolio (HJB) appears to give smoother equity growth compared to the Tangent Portfolio.

More testing is in progress, and the models will be refined further.

Figures to show (optional):

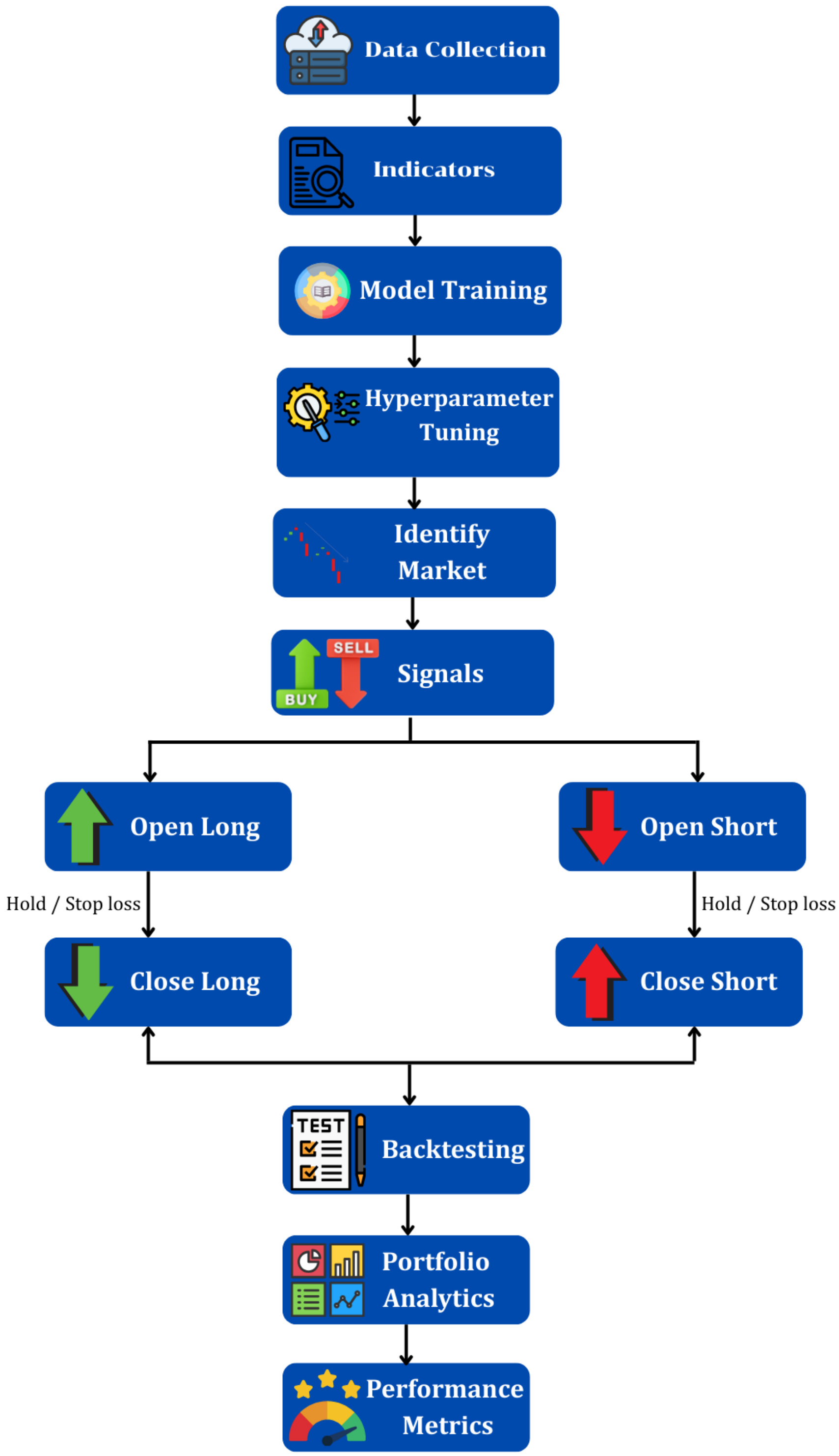
- Equity curve chart.
- Return vs Volatility scatter plot.
- Table of results.

4. Discussion

Good points:

- End-to-end process (data -> model -> signals -> backtest -> portfolio).
- Mix of AI and finance methods.
- Include transaction cost for more realistic test.

Methodology Diagram



Figure

```
WARNING:tvDatafeed.main:you are using nologin method, data you access may be limited
Using GPU
Fetching data for ['XAUUSD', 'XAGUSD', 'ZW1!', 'AOT', 'BRN1!', 'S501!'] using tvDatafeed...
Fetching XAUUSD (Gold Spot) from OANDA ...
OK: XAUUSD, rows=245
Fetching XAGUSD (Silver Spot) from OANDA ...
OK: XAGUSD, rows=245
Fetching ZW1! (Wheat Futures) from CBOT ...
OK: ZW1!, rows=255
Fetching AOT (AOT) from SET ...
OK: AOT, rows=284
Fetching BRN1! (Brent Futures) from ICEEUR ...
OK: BRN1!, rows=245
Fetching S501! (SET50 Futures) from TFEX ...
```

Figure 1. Example of fetching historical market data using *tvDatafeed*.

Assets include gold (XAUUSD), silver (XAGUSD), wheat futures (ZW1!), Thai stock (AOT), Brent oil futures (BRN1!), and SET50 index futures (S501!).

```
[before tuning] {'sequence_length': 38, 'hidden_layer_size': 492, 'lstm_layers': 1, 'dropout': 0.0, 'learning_rate': 0.0021, 'epochs': 3000, 'tx_cost_bps': 10.0, 'long_threshold': 0.0, 'short_threshold': 0.0, 'train_frac': 0.7}
c:\Users\offen\AppData\Local\Programs\Python\Python312\Lib\site-packages\optuna\distributions.py:708: UserWarning: The distribution is specified by [21, 91] and step=4, but the range is not divisible by `step`. It will be replaced by
  warnings.warn(
[I 2025-08-23 16:49:08,031] Trial 0 finished with value: 0.2844489602178868 and parameters: {'sequence_length': 33, 'hidden_layer_size': 1256, 'lstm_layers': 2, 'dropout': 0.4712151502282615, 'learning_rate': 0.007675507817190953, 'ep
c:\Users\offen\AppData\Local\Programs\Python\Python312\Lib\site-packages\optuna\distributions.py:708: UserWarning: The distribution is specified by [21, 91] and step=4, but the range is not divisible by `step`. It will be replaced by
  warnings.warn(
[I 2025-08-23 16:49:10,856] Trial 1 finished with value: -2.540892391084415 and parameters: {'sequence_length': 89, 'hidden_layer_size': 1760, 'lstm_layers': 2, 'dropout': 0.3005970753140752, 'learning_rate': 0.003373718916756485 3, 'e
c:\Users\offen\AppData\Local\Programs\Python\Python312\Lib\site-packages\optuna\distributions.py:708: UserWarning: The distribution is specified by [21, 91] and step=4, but the range is not divisible by `step`. It will be replaced by
  warnings.warn(
[I 2025-08-23 16:49:11,075] Trial 2 finished with value: 0.0 and parameters: {'sequence_length': 57, 'hidden_layer_size': 56, 'lstm_layers': 3, 'dropout': 0.5295847143816699, 'learning_rate': 0.0002237187115484708, 'epochs': 3080, 'lo
c:\Users\offen\AppData\Local\Programs\Python\Python312\Lib\site-packages\optuna\distributions.py:708: UserWarning: The distribution is specified by [21, 91] and step=4, but the range is not divisible by `step`. It will be replaced by
  warnings.warn(
[I 2025-08-23 16:49:12,693] Trial 3 finished with value: 0.0 and parameters: {'sequence_length': 85, 'hidden_layer_size': 1312, 'lstm_layers': 2, 'dropout': 0.47323808576444726, 'learning_rate': 0.0001485823805383844, 'epochs': 2850,
c:\Users\offen\AppData\Local\Programs\Python\Python312\Lib\site-packages\optuna\distributions.py:708: UserWarning: The distribution is specified by [21, 91] and step=4, but the range is not divisible by `step`. It will be replaced by
  warnings.warn(
[I 2025-08-23 16:49:14,015] Trial 4 finished with value: 0.0 and parameters: {'sequence_length': 77, 'hidden_layer_size': 312, 'lstm_layers': 3, 'dropout': 0.4227487849137435, 'learning_rate': 6.446224643080463e-05, 'epochs': 4630, 'l
c:\Users\offen\AppData\Local\Programs\Python\Python312\Lib\site-packages\optuna\distributions.py:708: UserWarning: The distribution is specified by [21, 91] and step=4, but the range is not divisible by `step`. It will be replaced by
  warnings.warn(
[I 2025-08-23 16:49:14,145] Trial 5 finished with value: 0.06874385365138735 and parameters: {'sequence_length': 25, 'hidden_layer_size': 392, 'lstm_layers': 1, 'dropout': 0.4049285661493981, 'learning_rate': 0.001583066741032151 2, 'e
c:\Users\offen\AppData\Local\Programs\Python\Python312\Lib\site-packages\optuna\distributions.py:708: UserWarning: The distribution is specified by [21, 91] and step=4, but the range is not divisible by `step`. It will be replaced by
  warnings.warn(
[I 2025-08-23 16:49:14,365] Trial 6 finished with value: 0.05197209475822362 and parameters: {'sequence_length': 41, 'hidden_layer_size': 1024, 'lstm_layers': 1, 'dropout': 0.36431622373109074, 'learning_rate': 0.0012399730859723546,
c:\Users\offen\AppData\Local\Programs\Python\Python312\Lib\site-packages\optuna\distributions.py:708: UserWarning: The distribution is specified by [21, 91] and step=4, but the range is not divisible by `step`. It will be replaced by
  warnings.warn(
[I 2025-08-23 16:49:19,640] Trial 7 finished with value: 0.0 and parameters: {'sequence_length': 77, 'hidden_layer_size': 1992, 'lstm_layers': 3, 'dropout': 0.47517848117498385, 'learning_rate': 0.00011353630204166286, 'epochs': 3130,
c:\Users\offen\AppData\Local\Programs\Python\Python312\Lib\site-packages\optuna\distributions.py:708: UserWarning: The distribution is specified by [21, 91] and step=4, but the range is not divisible by `step`. It will be replaced by
...
[I 2025-08-23 17:01:55,650] Trial 998 finished with value: 0.02096629336867384 and parameters: {'sequence_length': 41, 'hidden_layer_size': 560, 'lstm_layers': 3, 'dropout': 0.2354522987635267, 'learning_rate': 0.00766698617728887 9, '
c:\Users\offen\AppData\Local\Programs\Python\Python312\Lib\site-packages\optuna\distributions.py:708: UserWarning: The distribution is specified by [21, 91] and step=4, but the range is not divisible by `step`. It will be replaced by
  warnings.warn(
[I 2025-08-23 17:01:56,127] Trial 999 finished with value: 0.36897337884587156 and parameters: {'sequence_length': 37, 'hidden_layer_size': 688, 'lstm_layers': 3, 'dropout': 0.5630134794079652, 'learning_rate': 0.005268854051770085 5,
```

Figure 2. Example of Optuna hyperparameter tuning log.

Each line shows the trial number, objective value, and selected parameters. In this project, tuning was used to search for the best sequence length, hidden size, number of layers, dropout rate, learning rate, and epochs to optimize model performance.

```
=== Best Trial ===
Value (objective): 1.479902858207205
sequence_length: 37
hidden_layer_size: 800
lstm_layers: 3
dropout: 0.5781920902236091
learning_rate: 0.004788948670936635
epochs: 610
long_threshold: 0.0025283114114456233
short_threshold: 0.0023682367050814135
Metrics (val): {'Total Profit %': np.float64(2.659710628275902), 'MaxDD %': -0.29495194251717427, 'WinRate': 0.36}
[after tuning] {'sequence_length': 37, 'hidden_layer_size': 800, 'lstm_layers': 3, 'dropout': 0.5781920902236091, 'learning_rate': 0.004788948670936635,
```

Figure 3. Best Trial from Hyperparameter Tuning (Optuna)

The output shows the best hyperparameters found during Optuna tuning for the LSTM model. Parameters such as sequence length, hidden layer size, number of layers, dropout, learning rate, and thresholds are optimized. The best configuration achieved a profit of +2.65% with low drawdown (−0.29%).

```
Processing XAUUSD ...
[params for XAUUSD ] {'sequence_length': 37, 'hidden_layer_size': 800, 'lstm_layers': 3, 'dropout': 0.5781920902236091, 'learning_rate': 0.004788948670936635, 'epochs': 610, 'long_threshold': 0.0025283114114456233,
[train] seq=37 | hidden=800 | layers=3 | dropout=0.58 | lr=0.004789 | epochs=610 | thr=[long 0.0025, short 0.0024]
Epoch 50/610  TrainLoss: 0.03672156
Epoch 100/610  TrainLoss: 0.03693374
Epoch 150/610  TrainLoss: 0.03634100
Epoch 200/610  TrainLoss: 0.03654732
Epoch 250/610  TrainLoss: 0.03644609
Epoch 300/610  TrainLoss: 0.03475150
Epoch 350/610  TrainLoss: 0.03708975
Epoch 400/610  TrainLoss: 0.03689101
...
Epoch 500/610  TrainLoss: 0.03675702
Epoch 550/610  TrainLoss: 0.03670326
Epoch 600/610  TrainLoss: 0.03679040
Epoch 610/610  TrainLoss: 0.03676088
```

Figure 4. Training Log for XAUUSD (Gold)

The log displays the training process of the LSTM model on XAUUSD with the best parameters from tuning. The model runs for 610 epochs, and the training loss converges around 0.036, showing stable learning of price dynamics.

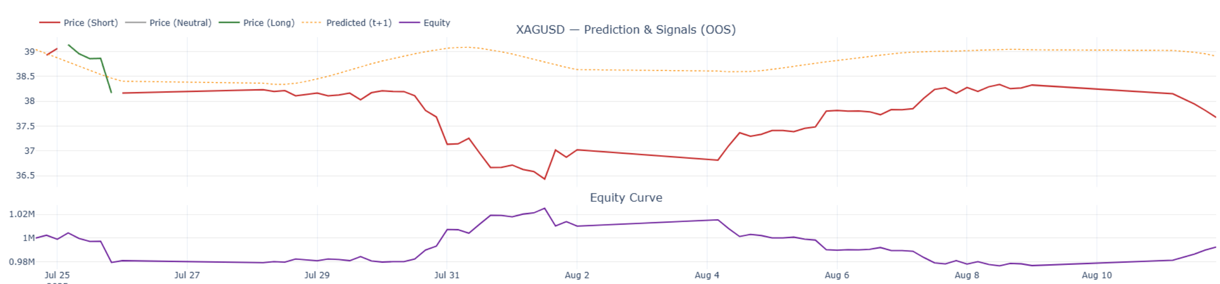


Figure 5. Example of trading simulation with machine learning (XAGUSD).

The top panel shows trading signals: green markers indicate long entries and red markers indicate short entries.
The **dashed line** represents the predicted next-step price.
The **bottom panel** shows the equity curve, starting from the initial investment balance.

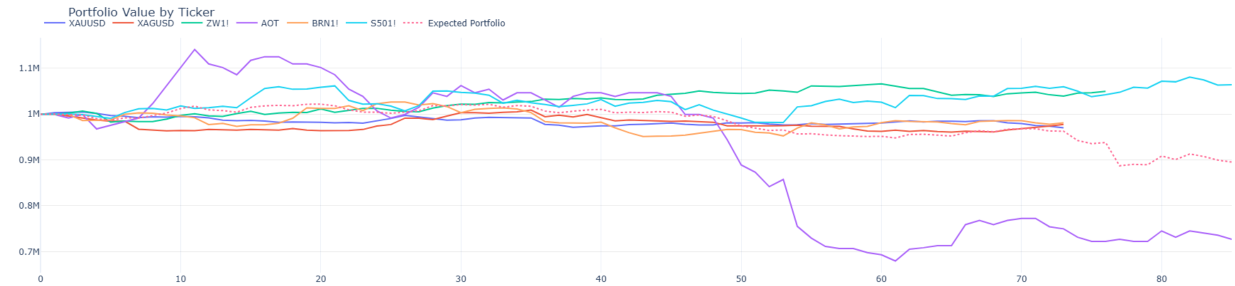


Figure 6. Portfolio Value by Ticker

This chart shows how the portfolio value changes over time for each asset: XAUUSD (gold), XAGUSD (silver), ZW1! (wheat), AOT, BRN1! (brent), and S501! (SET50 futures). The solid lines represent individual asset portfolios, while the dashed cyan line shows the expected (average) portfolio. Some assets, like ZW1! and S501!, perform steadily, while others such as AOT show a significant drop. This figure helps compare risk and return behavior across different assets.

Performance Summary

Asset	Final Portfolio Value (Cash)	Total Profit (Cash)	Total Profit (Percentage)	Win Rate	Average Profit (Percentage)	Average Loss (Percentage)	Max Drawdown (Percentage)	Sigma (Period % stdev)	Num Trades
XAUUSD	959,455.32	-40,544.68	-4.05%	0.00%	0.00%	-1.35%	-4.11%	0.30%	3
XAGUSD	992,438.34	-7,561.66	-0.76%	66.67%	0.65%	-2.06%	-4.76%	0.45%	3
ZW1!	1,045,224.22	45,224.22	4.52%	100.00%	4.52%	0.00%	-2.61%	0.47%	1
AOT	757,029.17	-242,970.83	-24.30%	25.00%	1.28%	-8.53%	-36.25%	2.41%	4
BRN1!	993,097.93	-6,902.07	-0.69%	20.00%	2.78%	-0.87%	-5.71%	0.71%	5
S501!	988,971.78	-11,028.22	-1.10%	50.00%	4.29%	-5.39%	-13.85%	1.25%	2

Figure 7. Performance Summary Table

This table reports the backtesting results of each asset. It shows portfolio value, profit/loss (cash and percentage), win rate, average profit/loss, maximum drawdown, volatility (σ), and number of trades. For example, wheat futures (ZW1!) produced a +4.52% return with 100% win rate, while AOT experienced a -24.30% loss with large drawdown.

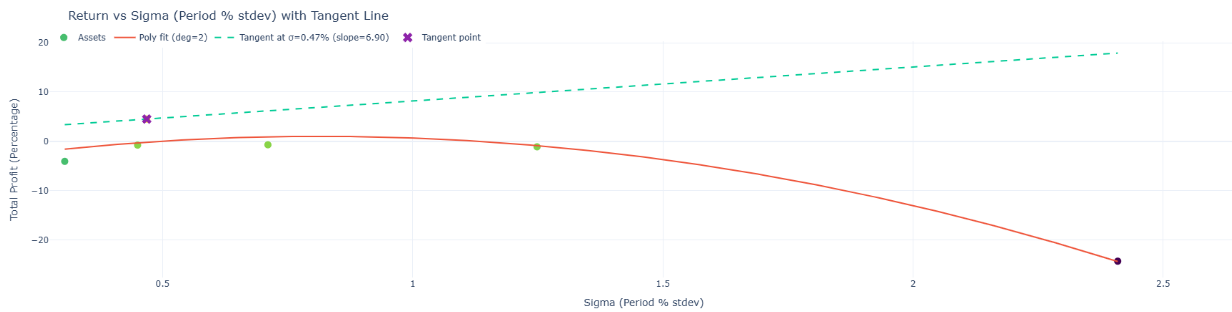
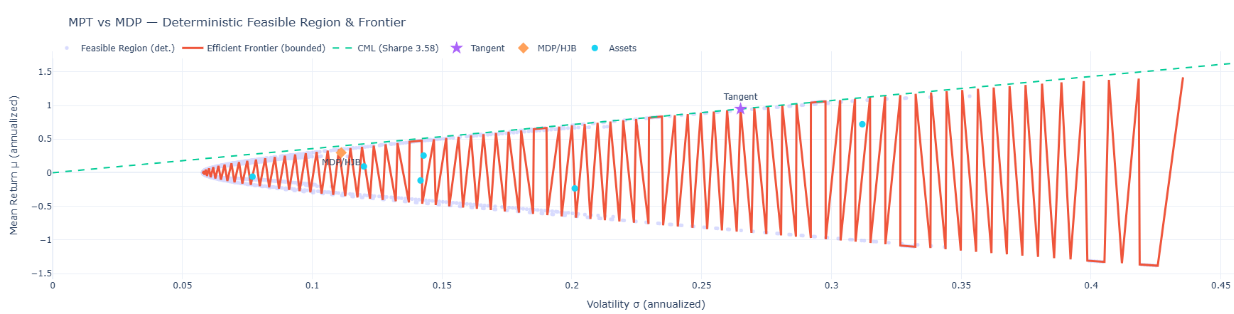


Figure 8. Return vs Sigma with Tangent Line

The scatter plot shows the relationship between return (%) and volatility (σ) for individual assets. A polynomial fit (red curve) represents the general trend, while the tangent line (cyan) indicates the optimal trade-off between risk and return. The tangent point (purple X) highlights the portfolio with the highest Sharpe ratio.




```
Tangent weights: {'XAUUSD': np.float64(0.0),
Final equity (tangent): 1129895.975396731
Final equity (dynamic): 1133624.587185193
```

Figure 9. Efficient Frontier: MPT vs MDP (HJB)

This figure compares Modern Portfolio Theory (MPT) with the Merton Dynamic Portfolio (MDP/HJB). The red curve shows the efficient frontier from static portfolios, while the dashed cyan line represents the Capital Market Line (CML). The tangent portfolio (purple star) and the MDP/HJB solution (blue dot) illustrate different approaches to achieving optimal portfolios.