Summary of the Paper: Deep Neural Networks, Gradient-Boosted Trees, Random Forests: Statistical Arbitrage on the S&P 500

Stock Universe Traded

- **S&P 500**: The paper focuses on the 500 leading companies in the U.S. stock market, which account for approximately 80% of the available market capitalization.

Data Period

- **Timeframe**: December 1992 to October 2015.

Research Sample

- **In-Sample Data**: The in-sample training window is set to 750 days (approximately three years) for training the models.
- **Out-of-Sample Data**: The out-of-sample trading window is set to 250 days (approximately one year) for applying the models. The dataset is divided into 23 non-overlapping training-trading batches.

Methodology

1. **Feature Generation**:
   - **Inputs**: Lagged returns of S&P 500 stocks, considering multi-period returns ($m \in \{1, 2, ..., 20, 40, 60, ..., 240\}$).
   - **Outputs**: Binary response variable indicating whether a stock's one-period return is greater than the median return of all stocks.
2. **Models Used**:
   - **Deep Neural Networks (DNNs)**:
     - Architecture: 31-31-10-5-2 layers.
     - Training: 400 epochs with dropout regularization and ADADELTA optimization.
   - **Gradient-Boosted Trees (GBTs)**:
     - Parameters: 100 trees, tree depth of 3, learning rate of 0.1, and 15 randomly selected features at each split.
   - **Random Forests (RAFs)**:
     - Parameters: 1000 trees, tree depth of 20, and approximately $\sqrt{p}$ features at each split.
3. **Ensemble Model (ENS)**:
   - Combination of the three models by averaging their probability forecasts.
4. **Trading Strategy**:
   - Daily one-day-ahead trading signals are generated based on the probability forecasts.
   - **Top k** probabilities are converted into long positions, and the **lowest k** into short positions, excluding the middle part of the ranking to reduce uncertainty.

Results

- **Returns**:

  - The ensemble model produced out-of-sample returns exceeding 0.45% per day for k=10 before transaction costs.
  - After transaction costs, returns reduced to 0.25% per day.
- **Performance Metrics**:
  - **Sharpe Ratio**: The ensemble strategy achieved a Sharpe ratio of 1.81, significantly higher than the general market.
  - **Sub-period Performance**: The strategy performed exceptionally well during the dot-com bubble and the global financial crisis, but returns deteriorated in recent years due to increasing market efficiency and popularization of machine learning techniques.

Summary of "Statistical Arbitrage in the U.S. Equities Market"

Stock Universe

The study focuses on a broad universe of U.S. equities. Specifically, the stocks in the universe have a market capitalization of more than 1 billion USD at the trade date.

Data Period

The research covers the period from 1997 to 2007. Specific back-testing of ETF strategies includes data from 2002 to 2007 due to the availability of actual ETFs.

Research Sample

- In-sample data: Used for parameter estimation and model development, consisting of historical data prior to the trade date.
- Out-of-sample data: Used for back-testing, where estimation of the residual process at time \(t \) uses only information available before this time, ensuring no look-ahead bias.

Methodology and Features

Principal Component Analysis (PCA) Approach

- Stocks' returns are decomposed into systematic and idiosyncratic components using PCA.
- PCA extracts risk-factors from the data, focusing on the significant eigenvalues that explain the majority of the variance.
- The residuals (idiosyncratic components) are modeled as mean-reverting processes.
- A fixed number of PCA factors (e.g., 15) or a variable number of factors explaining a certain percentage (e.g., 55\%) of the total variance are used to

generate trading signals.

ETF-Based Approach

- Uses sector Exchange Traded Funds (ETFs) as proxies for risk-factors.
- A regression of stock returns on the corresponding ETF returns generates residuals.
- The residuals are modeled similarly as mean-reverting processes.
- An adjustment for trading volume is introduced, considering signals in "trading time" to account for the impact of trading volume on the reliability of the signals.

Features of the Trading Strategy

The primary focus is on mean-reversion, generating contrarian trading signals. Trades are initiated when the residual deviates significantly from its mean (measured by the s-score). Trading signals are:

- Buy to open if $s_i < -s_{bo}$
- Sell to open if $s_i > +s_{so}$
- Close short position if $s_i < +s_{bc}$
- Close long position if $s_i > -s_{sc}$

S-scores are calibrated empirically, with suggested cutoffs $s_{bo} = s_{so} = 1.25$, $s_{bc} = 0.75$, and $s_{sc} = 0.50$.

Back-Testing Results

PCA-Based Strategies

- Average annual Sharpe ratio of 1.44 over 1997-2007.
- Performance degradation observed post-2003, with an average Sharpe ratio of 0.9 during 2003-2007.

ETF-Based Strategies

- Achieved a Sharpe ratio of 1.1 from 1997 to 2007.
- Improved performance (Sharpe ratio of 1.51) from 2003 to 2007 when incorporating trading volume information.

Performance During Market Events

The study also examines the performance during the liquidity crisis of August 2007. The results are consistent with the "unwinding" theory of Khandani and Lo (2007), where simultaneous exits from positions by multiple funds caused significant market movements.

Conclusion

In conclusion, the paper presents a comprehensive analysis of statistical arbitrage strategies in the U.S. equities market, focusing on mean-reversion and the efficacy of PCA and ETF-based approaches. The integration of trading volume information is highlighted as a significant improvement for ETF-based strategies.

Summary of the Paper: "Empirical Asset Pricing via Machine Learning"

Stock Universe and Data Period

- **Stock Universe**: The study includes nearly 30,000 individual stocks.
- **Data Period**: The data covers 60 years, from 1957 to 2016.

Research Sample

- **In-Sample Data**: The in-sample period is used for training the models. This part of the data is employed to fit the models and fine-tune hyperparameters using various machine learning techniques.
- **Out-of-Sample Data**: The out-of-sample period is used for validation and testing. This part of the data is employed to evaluate the model's performance in making predictions, ensuring the robustness and generalizability of the findings.

Methodology

The study employs various machine learning methods to predict stock returns and measure asset risk premiums. The primary methods and features are as follows:

1. **Linear Models**:
    - Simple Linear Regression
    - Penalized Linear Models (Elastic Net combining Lasso and Ridge Regression)
    - Dimension Reduction Techniques (Principal Components Regression, Partial Least Squares)
2. **Nonlinear Models**:
    - Generalized Linear Models (using spline functions for nonlinearity)
    - Regression Trees and Ensembles (Boosted Regression Trees, Random Forests)
    - Neural Networks (various architectures from shallow to deep networks)

Features Used

The models use an extensive set of predictor variables, totaling over 900 baseline signals, which include:

- **94 stock characteristics**: Such as size, book-to-market ratio, momentum, etc.
- **Interactions with 8 aggregate time-series variables**: Macro-economic factors that influence the stock market.
- **74 industry sector dummy variables**: To account for sector-specific effects.

The study also highlights the most successful predictors:

- **Price Trends**: Stock momentum, industry momentum, short-term reversal.
- **Liquidity Variables**: Market value, dollar volume, bid-ask spread.
- **Volatility Measures**: Return volatility, idiosyncratic volatility, market beta, beta squared.

Key Findings

- **Predictive Accuracy**: Machine learning methods significantly improve predictive accuracy over traditional regression-based strategies. Tree-based methods and neural networks provide the best performance due to their ability to capture complex nonlinear interactions.
- **Economic Gains**: Portfolio strategies based on machine learning forecasts yield substantial economic gains. For instance, neural network forecasts for the S&P 500 result in a much higher out-of-sample Sharpe ratio compared to a buy-and-hold strategy.
- **Feature Importance**: Despite the variety of models, all methods consistently identify the same set of dominant predictive signals, emphasizing the robustness of the findings.

Summary of the Paper: "Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions"

Stock Universe and Data Period

- **Stock Universe**: The study focuses on the constituent stocks of the S&P 500 index.
- **Data Period**: The data covers the period from 1992 to 2015.

Research Sample

- **In-Sample Data**: The training set consists of 750 days (approximately three years) of historical data.
- **Out-of-Sample Data**: The trading set consists of 250 days (approximately one year) of data used for out-of-sample predictions.

Methodology

1. **Data Preprocessing**:
   - **Feature Generation**: Daily returns are calculated and standardized. The study uses sequences of 240 consecutive, standardized one-day returns as input features for the LSTM networks.
   - **Target Generation**: A binary classification problem is set up where stocks are classified based on whether their return is above or below the cross-sectional median return for the next day.
2. **Model Architecture**:
   - **LSTM Networks**: These networks consist of an input layer, one or more hidden LSTM layers, and an output layer. The hidden layers

    include memory cells with input, forget, and output gates that manage cell states and outputs.
  - **Training**: The LSTM model is trained using the RMSprop optimizer, dropout regularization, and early stopping to prevent overfitting. The training process uses 80% of the data as the training set and 20% as the validation set.
3. **Benchmark Models**:
   - **Random Forests (RAF)**: An ensemble learning method using multiple decision trees.
   - **Deep Neural Networks (DNN)**: A standard feedforward neural network with multiple hidden layers.
   - **Logistic Regression (LOG)**: A baseline classifier using L2 regularization.
4. **Trading Strategy**:
   - Stocks are ranked based on the predicted probabilities of outperforming the cross-sectional median return. Long positions are taken in the top k stocks, and short positions are taken in the bottom k stocks, forming a long-short portfolio.

Performance Evaluation

- **Daily Returns**:
  - LSTM: 0.46% before transaction costs, 0.26% after transaction costs.
  - RAF: 0.43% before transaction costs, 0.23% after transaction costs.
  - DNN: 0.32% before transaction costs, 0.12% after transaction costs.
  - LOG: 0.26% before transaction costs, 0.06% after transaction costs.
- **Sharpe Ratio**:
  - LSTM: 5.83 before transaction costs, 2.34 after transaction costs.
  - RAF: 5.00 before transaction costs, 1.87 after transaction costs.
  - DNN: 2.43 before transaction costs, 0.52 after transaction costs.
  - LOG: 1.70 before transaction costs, 0.10 after transaction costs.
- **Accuracy**:
  - LSTM: 54.3% correct classifications.
  - RAF: 52.6% correct classifications.
  - DNN: 52.4% correct classifications.
  - LOG: 51.1% correct classifications.

Key Findings

1. **Predictive Performance**: LSTM networks outperform other memory-free classification methods (RAF, DNN, LOG) in terms of both predictive accuracy and economic performance.
2. **Sources of Profitability**: Stocks selected for trading by the LSTM network exhibit high volatility and a short-term reversal return profile.
3. **Simplified Trading Strategy**: A rules-based short-term reversal strategy

that buys short-term extremal losers and sells short-term extremal winners explains a portion of the returns achieved by the LSTM-based strategy, indicating the LSTM's ability to capture complex patterns.

Conclusion

The study demonstrates the effectiveness of LSTM networks in predicting financial market movements and constructing profitable trading strategies. The LSTM-based approach significantly outperforms traditional models and reveals underlying patterns that contribute to its profitability.

Stock Universe
- **Traded Stocks**: The study focuses on the constituent stocks of the S&P 500 index.
- **Source**: Data for these stocks is sourced from Thomson Reuters.

Data Period
- **Time Frame**: The data period spans from January 1990 to October 2015.
- **Survivor Bias Elimination**: The authors obtain all month-end constituent lists for the S&P 500 from December 1989 to September 2015 to eliminate survivor bias, allowing them to reproduce the S&P 500 index composition at any given time within this period.

Research Sample
- **In-Sample Data**: Training periods consist of 750 days (approximately three years).
- **Out-of-Sample Data**: Trading periods consist of 250 days (approximately one year).
- **Study Periods**: The entire data set from 1990 to 2015 is split into 23 non-overlapping study periods, each comprising one training and one trading period.
- **Stocks Included**: All stocks that were constituents of the S&P 500 at the end of each training period are included in both training and trading sets.

Methodology
- **Model**: Long Short-Term Memory (LSTM) networks are used for predicting out-of-sample directional movements.
- **Benchmark Models**: The performance of the LSTM networks is compared against Random Forests (RAF), Deep Neural Networks (DNN), and Logistic Regression (LOG).
- **Features**: Standardized one-day returns are used as the primary feature for LSTM networks. For other models, cumulative returns over multiple periods are used.
- **Training and Trading**: The LSTM is trained on sequences of standardized one-day returns over 240 days. Predictions are made for the next day’s return

direction relative to the cross-sectional median.

Features Used
- **Return Sequences**: Standardized one-day returns are used for LSTM networks, calculated as:

\[
R_{m,s,t} = \frac{P^s_t}{P^s_{t-m}} - 1
\]

where \(R_{m,s,t} \) is the simple return for stock \(s \) over \(m \) periods.
- **Standardization**: Returns are standardized by subtracting the mean and dividing by the standard deviation from the training set:

\[
\tilde{R}_{m,s,t} = \frac{R_{m,s,t} - \mu_{m,\text{train}}}{\sigma_{m,\text{train}}}
\]

- **Input Sequences**: Sequences of 240 consecutive standardized one-day returns are generated for training:

\[
\{\tilde{R}_{1,s,t-239}, \tilde{R}_{1,s,t-238}, \ldots, \tilde{R}_{1,s,t}\}
\]

- **Targets**: A binary classification problem is defined, where the response variable indicates whether the one-period return is above or below the median return of all stocks.

Performance
- **Daily Returns**: LSTM networks achieved daily returns of 0.46% prior to transaction costs.
- **Sharpe Ratio**: LSTM networks achieved a Sharpe ratio of 5.8 prior to transaction costs.
- **Comparative Performance**: LSTM networks outperformed RAF, DNN, and LOG in terms of daily returns and Sharpe ratio.

Findings
- **Volatility and Reversal**: Stocks selected for trading by the LSTM network exhibited high volatility and a short-term reversal return profile.
- **Simplified Strategy**: A rules-based short-term reversal strategy explained a portion of the LSTM returns, but the LSTM network's superior performance was attributed to its ability to capture more complex patterns in the return sequences.

"Deep Learning Statistical Arbitrage" by Jorge Guijarro-Ordonez, Markus Pelger, and Greg Zanotti:

Stock Universe
- **Traded Stocks**: The study focuses on the largest and most liquid stocks in the U.S., roughly corresponding to the S&P 500 index.
- **Source**: Data for these stocks is sourced from CRSP and supplemented with firm-specific characteristics from the CRSP/Compustat database.

Data Period
- **Time Frame**: The data period spans from January 1998 to December 2016.
- **Initial Estimation Period**: Data from January 1978 to December 1997 is used for initial factor model estimation.
- **Rolling Window Estimation**: Factor models are estimated using a rolling window approach to ensure out-of-sample performance.

Research Sample
- **In-Sample Data**: Daily returns from 1998 to 2016, with initial estimation from 1978 to 1997.
- **Out-of-Sample Data**: The model is trained on a rolling window of 1,000 days and re-estimated every 125 days.
- **Stocks Included**: The largest 550 stocks by market capitalization, which account for 0.01% of the total market capitalization.

Methodology
- **Model**: A convolutional neural network (CNN) combined with a transformer network is used for predicting time-series signals and generating optimal trading policies.
- **Benchmark Models**: The performance of the deep learning model is compared against parametric models like Ornstein-Uhlenbeck (OU) and Fourier transformation with feedforward neural networks (FFN).
- **Factor Models**: Three classes of factor models are used:
  1. **Fama-French Factors**: Includes CAPM, Fama-French 3, 5, and 8 factor models.
  2. **PCA Factors**: Includes 1, 3, 5, 8, 10, and 15 latent factors.
  3. **IPCA Factors**: Includes 1, 3, 5, 8, 10, and 15 factors with conditional latent factors based on firm-specific characteristics.

Features Used

- **Residuals Calculation**: Residuals are computed relative to the factor models.
- **Input Sequences**: Sequences of the last 30 lagged residuals are used for signal extraction.
- **Convolutional Filters**: Local filters identify patterns in the data, with a flexible time-series model capturing complex dependencies.
- **Attention Mechanism**: Transformers use attention mechanisms to model global patterns in the time-series data.

Performance
- **Sharpe Ratio**: The deep learning model achieves a Sharpe ratio larger than four.
- **Annual Returns**: The model can achieve annual out-of-sample mean returns of 20%.
- **Comparison**: The deep learning model outperforms all benchmark approaches, including traditional parametric and alternative deep learning models without convolutional transformers.

Findings
- **Arbitrage Opportunities**: The model identifies substantial short-term arbitrage opportunities in financial markets.
- **Signal Importance**: The choice of signal extraction method is crucial, with the convolutional transformer significantly improving performance over traditional filters.
- **Trading Strategies**: The model suggests trading strategies that are robust to transaction costs and realistic market frictions.

Methodology Details
1. **Arbitrage Portfolio Generation**: Portfolios are generated as residuals from conditional latent asset pricing factors.
2. **Arbitrage Signal Extraction**: Signals are extracted using a convolutional transformer, which combines local pattern recognition with global temporal dependencies.
3. **Trading Policy**: The extracted signals are used to form an optimal trading policy that maximizes risk-adjusted returns under constraints.