Time-series based pattern classification in finance

Nandhini Rajasekaran
Systems Design Department, University of Waterloo, Canada
{ n2rajase }@uwaterloo

***Abstract*—Financial markets are complex, dynamic systems marked by high volatility, nonlinearity, and rapid shifts driven by internal dynamics and external events. These characteristics pose significant challenges for accurate market trend prediction and reliable trading strategy development, as traditional forecasting models often struggle with noise, abrupt changes, and intricate dependencies in financial time series data.**

This study proposes a novel approach that integrates traditional time series analysis, deep learning, wavelet-based features, and attention mechanisms into a recurrent neural network (RNN) to enhance market trend prediction and trading performance. The model is trained in a simulated trading environment using experience replay and an epsilon-greedy strategy, enabling the agent to learn from historical data and adapt its decision-making over time. Empirical evaluations demonstrate that the proposed model achieves improved reward consistency and trend classification accuracy, highlighting the efficacy of hybrid temporal frequency representation learning for financial decision-making. Furthermore, the model's ability to capture multi-scale temporal patterns and prioritize relevant features contributes to its robustness across diverse market conditions. These findings suggest potential applications in automated trading systems and portfolio optimization.

***Index Terms*—LSTMs, GRUs, Wavelets, Deep-Q-Network, and PPO Agent.**

I. Introduction

Financial markets are dynamic arenas where prices fluctuate rapidly due to economic indicators, investor behavior, and global events. Time series pattern classification in these markets is both critical and challenging, as market data exhibit volatility, nonlinearity, and non-stationarity, often obscured by noise and abrupt shifts. Unlike systems governed by predictable rules, financial markets are shaped by a mix of quantitative metrics and qualitative factors, such as news, earnings reports, and market sentiment, making accurate trend prediction a formidable task.

The ability to identify patterns in financial time series—whether directional trends, turning points, or volatility shifts—drives critical applications like algorithmic trading, risk management, and portfolio optimization. For instance, classifying market conditions as bullish or bearish can inform timely "Buy," "Sell," or "Hold" decisions, while anticipating regime changes can mitigate losses. However, traditional approaches, such as statistical models or rule-based systems, often fail to capture the contextual nuances and sequential dependencies inherent in financial data.

To address these challenges, this study leverages historical S&P 500 Index (GSPC) data, sourced through the Yahoo Finance API, as a robust benchmark for evaluating advanced classification models. The S&P 500, representing a broad cross section of US. equities, serves as a reliable proxy for market behavior and investor sentiment. Our research introduces a scalable, context-aware learning framework designed to overcome the limitations of conventional methods. Drawing on insights from diverse domains, this approach innovatively combines multiscale temporal analysis and adaptive learning to classify complex market patterns with high accuracy and low latency. Its potential to power automated trading systems and dynamic investment strategies underscores its practical value. Through systematic experimentation, we evaluated its performance against established techniques, focusing on its ability to adapt to varying market regimes and deliver consistent results.

The following sections provide foundational concepts, detail our methodology and empirical results, discuss limitations, and propose avenues for future research.

II. Background

Forecasting financial market trends through time series analysis is a complex task due to the interplay of numerous factors influencing asset prices, including economic indicators, trading volume, and investor behavior. Time-series-based pattern classification is central to this endeavor, enabling the identification of trends, volatility patterns, and turning points that inform trading strategies and risk management. Over time, forecasting methods have evolved from linear statistical models to sophisticated deep learning and hybrid frameworks, driven by the need to handle market volatility, noise, and non-stationarity.

Early approaches to financial forecasting relied on statistical models like autoregressive Integrated Moving Average (ARIMA) and Generalized Autoregressive Conditional Heteroskedasticity (GARCH) to model trends and volatility. Kalman Filters, introduced by Kalman (1960) [1], were also widely used to smooth noisy market data and estimate latent states. While these models offer interpretability and perform well on stationary data, they struggle with the non-linearities and regime shifts common in financial markets, limiting their predictive power in dynamic environments.

The advent of greater computational resources led to the adoption of machine learning methods like Support Vector Machines (SVMs) and Random Forests, which capture nonlinear relationships in financial data. However, these methods require extensive feature engineering (e.g., constructing indicators like RSI or MACD) and lack the temporal awareness needed to model sequential dependencies effectively. As a result, their performance in real-time market forecasting remains suboptimal compared to sequence-aware models. Recurrent Neural

Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), marked a significant advancement in financial time series modeling. These architectures excel at capturing long-range temporal dependencies, making them well-suited for non-stationary market data. Fischer and Krauss (2018) [2] demonstrated that LSTMs outperform traditional models in predicting stock movements, while Bao et al. (2017) [3] combined LSTMs with denoising autoencoders to extract robust features from historical prices. GRUs, a computationally efficient variant of LSTMs, have shown comparable performance in financial applications, as noted in recent studies [4].

To address the challenge of noise and multi-scale patterns in financial data, Wavelet Transforms have gained prominence. Unlike Fourier Transforms, which assume stationarity, wavelets provide multi-resolution analysis, capturing both short-term fluctuations and long-term trends. Gencay et al. (2001, 2002) [5], [6] pioneered the use of wavelets in financial econometrics, showing their ability to enhance forecasting accuracy. Recent work by Zhang et al. (2019) [7] and Lin et al. (2021) [4] integrated wavelet-transformed signals with LSTM/GRU models, improving noise resilience and trend prediction in volatile markets.

Reinforcement Learning (RL) has emerged as a powerful approach for optimizing trading decisions through trial-and-error learning. Deep Q-Networks (DQN), as explored by Sutton and Barto (2018) [8], and experience replay techniques have been applied to develop adaptive trading strategies. These methods optimize action policies (e.g., Buy, Sell, Hold) by maximizing long-term rewards, complementing deep learning's feature extraction capabilities. Hybrid models combining LSTM-based classification with RL-driven decision-making, as in Wu et al. (2021) [9], have shown promise in balancing predictive accuracy and actionable trading outcomes.

Recent research has focused on hybrid models that integrate multiple techniques to improve generalization and robustness. For example, combining wavelet-based decomposition with deep learning enhances multi-scale pattern recognition, while attention mechanisms prioritize relevant temporal features, as seen in natural language processing and human activity recognition [10]. These cross-domain insights inspire our study, which adapts such techniques to financial trading. Unlike prior work, our hybrid framework emphasizes low-latency, real-time classification of nuanced market actions (Buy, Sell, Hold) under varying market regimes, addressing the challenge of ambiguous transitions during periods of uncertainty.

This study builds on these foundations by proposing a novel hybrid architecture that fuses wavelet-based time-frequency decomposition, LSTM-GRU sequence modeling, attention mechanisms, and RL with experience replay. By leveraging multi-scale temporal representations and context-aware decision optimization, our approach aims to overcome the limitations of prior models, offering improved accuracy and robustness for financial trading applications.

III. Method

A. *Dataset Preparation*

For this study, we utilized historical price data from GSPC, retrieved via the Yahoo Finance API using the `yfinance` Python library. The S&P 500, comprising 500 leading US publicly traded companies, represents approximately 80% of US market capitalization. Its broad sector diversity, high liquidity, and sensitivity to macroeconomic trends make it an ideal benchmark for evaluating time-series models in financial forecasting. The index's responsiveness to economic regimes such as bull markets, bear markets, and volatile periods enables robust analysis of market patterns and investor sentiment.

The dataset consists of daily price and volume data with the following fields:

- Open price
- High price
- Low price
- Close price
- Adjusted close price
- Volume
- Date (timestamp index)

To prepare the data for modeling, we applied several preprocessing steps to ensure consistency and compatibility with neural network training. Missing values were handled using forward-fill imputation to maintain temporal continuity. Timestamps were standardized to `datetime` format and sorted chronologically. Non-trading days, such as weekends and holidays, were excluded to eliminate irregular time gaps. Finally, numerical features (prices and volume) were normalized to the range $[0,1]$ using Min-Max scaling. This normalization ensures stable convergence during training and prevents features with larger magnitudes from disproportionately influencing the model.

We defined labels based on the percentage change in future prices over a forecast horizon of 3 days. The labeling rule is defined as follows:

$$\text{label}_t = \begin{cases} 0 & \text{if } \frac{P_{t+3}-P_t}{P_t} > \theta \quad \text{(UP)} \\ 1 & \text{if } \frac{P_{t+3}-P_t}{P_t} < -\theta \quad \text{(DOWN)} \\ 2 & \text{otherwise} \quad \text{(NEUTRAL)} \end{cases}$$

where $\theta = 0.005$ represents a 0.5% threshold for classifying significant movements. These trend labels were used for supervised classification and reinforcement learning reward design.

TABLE I
DATASET DISTRIBUTION

Label	Event	Number of Samples
0	Up	69
1	Down	119
2	Neutral	3450

B. *Feature Engineering*

To enhance the model's ability to capture complex market patterns, we engineered a comprehensive set of features from the dataset, combining time-domain and frequency-domain representations. Financial time series often exhibit high autocorrelation and multicollinearity, which can obscure subtle trends. By augmenting raw price and volume data with technical indicators and wavelet-based features, we provided the model with diverse perspectives on market behavior, capturing trend, momentum, and volatility dynamics.

The engineered feature set includes:

- **SMA-5, SMA-9, SMA-17:** Simple Moving Averages over 5, 9, and 17 days, capturing short- to medium-term trends. Shorter SMAs (e.g., SMA-5) respond quickly to price changes, while longer SMAs (e.g., SMA-17) smooth out noise.
- **SMA Crossover:** The difference between short- and long-term SMAs, signaling momentum shifts (e.g., bullish to bearish transitions).
- **RSI-14:** Relative Strength Index over 14 days, measuring the speed and magnitude of price movements to identify overbought or oversold conditions.
- **Stochastic Oscillator:** A 14-day momentum indicator comparing the current price to its recent range, highlighting potential reversals.
- **MACD:** Moving Average Convergence Divergence, derived from two EMAs, indicating trend direction and momentum.
- **MACD Signal:** A 9-day EMA of the MACD, used to trigger buy or sell signals.
- **ATR-14:** Average True Range over 14 days, quantifying market volatility to inform risk-adjusted decisions.
- **OBV:** On-Balance Volume, relating volume flow to price changes to detect momentum and potential divergences.
- **VWAP:** Volume Weighted Average Price, reflecting the average price weighted by volume, useful for identifying institutional trading levels.
- **Bollinger Bands:** Upper and lower bands based on a 20-day SMA and standard deviation, indicating price volatility and relative highs/lows.
- **BB Bandwidth:** The width between Bollinger Bands, signaling volatility squeezes or expansions.
- **Volatility:** Rolling standard deviation of returns over a 14-day window, representing short-term risk.
- **Volume:** Raw trading volume, retained to complement volume-based indicators.
- **Wavelet Transform:** Discrete Wavelet Transform (DWT) with the Daubechies-4 (`db4`) wavelet was applied to the closing price series, yielding approximation (A_1) and detail ($D_1, D_2, \ldots, D_n$) coefficients:

  $$x(t) \rightarrow [A_1, D_1, D_2, \ldots, D_n]$$

  where A_1 captures low-frequency trends and D_n represents high-frequency fluctuations. These coefficients were included as features.
- **Sliding Window:** Input sequences of length $T = 5$ days, containing normalized prices, technical indicators, and wavelet coefficients, were constructed to provide temporal context.

Feature selection was guided by financial intuition and statistical correlation analysis to ensure each feature contributes unique information. For example, SMA-based features (SMA-5, SMA-9, SMA-17, SMA Crossover) capture multi-horizon trends, with crossovers signaling critical turning points. ATR-14 and BB Bandwidth inform volatility-driven decisions, while OBV and VWAP provide volume-based insights into market momentum and institutional activity. The Wavelet Transform enhances the model's ability to discern local fluctuations and global trends, addressing the limitations of purely time-domain features. Combined, these features enable the model to classify nuanced market patterns, such as Buy, Sell, or Hold actions, under varying market regimes.

These feature engineering steps, building on the preprocessed dataset, were critical for robust model performance. By incorporating multi-dimensional market signals, we laid the foundation for effective training and evaluation of our hybrid learning framework.

C. *State Space Modeling*

State-space modeling provides a mathematical framework for representing dynamic systems where observed data are driven by latent states and noise. It comprises two equations: the state equation, which describes the evolution of the hidden state, and the observation equation, which links the state to observed variables. Formally, these are defined as:

$$x_t = Ax_{t-1} + w_t, \quad y_t = Hx_t + v_t,$$

where x_t is the hidden state (e.g., the underlying market trend), y_t is the observed data (e.g., closing price), A and H are the state transition and observation matrices, and w_t and v_t are Gaussian process and observation noise, respectively, with zero mean.

The Kalman Filter is a recursive algorithm that estimates the hidden state x_t from noisy observations y_t, assuming linear dynamics and Gaussian noise. It operates in two phases: a prediction phase, which forecasts the state based on the transition model, and an update phase, which refines the prediction using new observations. This process minimizes the mean squared error, providing an optimal state estimate.

In financial time series, the Kalman Filter excels at smoothing noisy price data to reveal latent trends obscured by market volatility. In this study, we applied the Kalman Filter to the closing prices to produce a denoised time series. This smoothed signal was incorporated as a feature in our hybrid model, enhancing its ability to detect directional trends and classify market patterns (e.g., Buy, Sell, Hold) under diverse regimes. By reducing short-term noise, the Kalman-filtered trend complements other features, such as wavelet coefficients and technical indicators, improving the model's robustness and trend prediction accuracy.

D. *Handling Class Imbalance*

Class imbalance presented a significant challenge in this study, with the NEUTRAL trend label dominating the dataset, while UP and DOWN labels were underrepresented. To address this, we evaluated conventional oversampling techniques, including **SMOTE** (Synthetic Minority Oversampling Technique) and **ADASYN** (Adaptive Synthetic Sampling). These methods generate synthetic samples for minority classes by interpolating feature-space similarities among existing examples.

However, SMOTE and ADASYN assume independent and identically distributed (i.i.d.) data, an assumption violated in financial time series where temporal ordering and sequential dependencies are critical. Applying these techniques to flattened feature windows disrupted sequence continuity, introducing unrealistic transitions between time steps. This resulted in negligible improvements in minority class performance and, in some cases, caused unstable training and overfitting to synthetic trends.

Our findings suggest that SMOTE and ADASYN are poorly suited for time series classification due to their disregard for temporal dependencies and yielded minimal improvements in precision and recall for UP and DOWN classes, often destabilizing training. In contrast, resample oversampling proved effective by maintaining sequence integrity, improving the model's ability to detect UP and DOWN trends. Architecture-level solutions, such as multi-head attention layers and loss reweighting, further enhanced performance by preserving sequential context. Future work could explore advanced time series-specific augmentation techniques, such as window warping or GAN-based synthesis, to address class imbalance in financial sequence modeling.

E. *Experimental Setup*

The preprocessed dataset was divided into training, validation, and testing subsets with an 80:10:10 split ratio, allocating 80% for training, 10% for validation, and 10% for testing. To mitigate overfitting and ensure robust model evaluation, we applied k-fold cross-validation with $k \in \{3, 5, 10\}$ during training, assessing performance across multiple data folds.

IV. Results & Analysis

Experiment 1: Time Series Forecasting with ARIMA + GARCH

This experiment evaluates the performance of a Autoregressive Integrated Moving Average combined with Generalized Autoregressive Conditional Heteroskedasticity model for forecasting trends. ARIMA models the conditional mean of the time series, capturing long-term trends, while GARCH accounts for time-varying volatility, enabling both point forecasts and confidence intervals.

As shown in Figure 1, the ARIMA component effectively captures the upward trend during the training period. However, its linear and stationary assumptions limit its ability to adapt to dynamic market regimes. In the test period, the model produces a conservative forecast, underestimating the observed bullish trend, with GARCH-derived confidence intervals ($\pm 1.96\sigma$) indicating volatility but lacking directional precision.

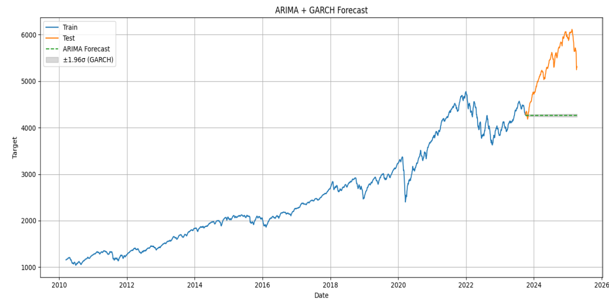

Fig. 1. ARIMA + GARCH forecast of S&P 500 trends, showing training (2009–2023) and testing (2024–2025) periods.

Observations:

- ARIMA struggles to extrapolate nonlinear trends, failing to capture sudden market regime shifts in the test period.
- GARCH confidence intervals widen during volatile periods but offer limited insight into trend directionality.
- The hybrid model performs adequately in stable markets but underperforms during strong bullish or bearish trends due to its linear constraints.

Experiment 2: LSTM Regression on Trend classification by Rule based system

This experiment evaluates a Long Short-Term Memory (LSTM) regression model for predicting future closing prices, with predictions converted to trend labels (UP, DOWN, NEUTRAL) via thresholding. The model was trained on sliding windows of historical price sequences and engineered features, outputting continuous price forecasts for a 3-day horizon.

Trend labels were assigned as:

$$\text{Label} = \begin{cases} \text{UP} & \text{if Change} > 0.5\% \\ \text{DOWN} & \text{if Change} < -0.5\% \\ \text{NEUTRAL} & \text{otherwise} \end{cases}$$

This approach enables interpretable trend classification while controlling sensitivity to minor price fluctuations.

Performance Metrics: The model's predictions were evaluated against true trend labels derived from actual price changes, yielding:

- **Overall Accuracy:** 74.6%
- **Macro-Average F1-Score:** 0.36
- **Weighted-Average F1-Score:** 0.73

Observations:

- High accuracy reflects strong performance on the dominant NEUTRAL class, driven by class imbalance.
- Low precision and recall for UP and DOWN classes indicate challenges in detecting directional shifts, exacerbated by the regression model's bias toward mean predictions.

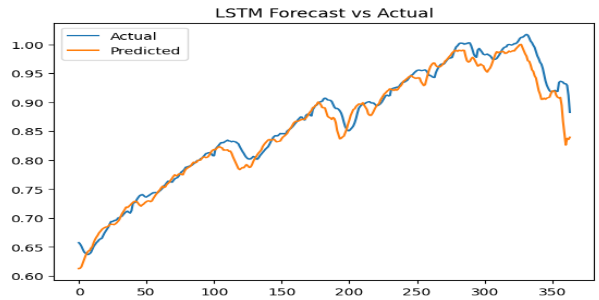

Fig. 2. LSTM regression with thresholding: Predicted vs. actual S&P 500 trends (2024–2025).

- The fixed thresholding strategy struggles with marginal price movements, suggesting the need for dynamic thresholds or direct classification approaches.
- Alternative strategies, such as reinforcement learning or attention-based classification, may better handle class imbalance and optimize trend-based decisions.

Experiment 3: Hybrid Model with Wavelet Features and Multi-Head Attention

This experiment evaluates a hybrid deep learning model for trend classification, integrating time-domain and frequency-domain features. The model combines normalized price data, engineered technical indicators, and Discrete Wavelet Transform (DWT) features to capture multi-scale patterns in non-stationary, noisy financial time series. Unlike Fourier transforms, DWT provides localized frequency information, decomposing the signal into:

$$x(t) \rightarrow [A_1, D_1, D_2, \ldots, D_n]$$

where A_1 represents low-frequency trends and D_n captures high-frequency volatility. These components enhance the model's ability to model both long-term trends and short-term fluctuations.

Model Architecture and Multi-Task Design: The model employs a sequence-to-sequence architecture with multi-task learning, jointly predicting future closing prices ($\hat{P}_{t+3}$) and trend labels (UP, DOWN, NEUTRAL). The architecture includes:

- **Input Layer:** Sliding window sequences of prices, technical indicators, and DWT components.
- **Stacked LSTM + GRU Layers:** Capture temporal dependencies across time steps.
- **Multi-Head Attention Layer:** Attends to diverse temporal features in parallel, emphasizing key patterns like volatility spikes or trend shifts.
- **Multi-Task Output Heads:** A regression head predicts $\hat{P}_{t+3}$, and a classification head assigns trend labels based on thresholded price changes.

This multi-task design optimizes shared temporal representations, improving generalization and efficiency.

Performance Metrics: Evaluated on the test set (2024–2025), the model achieved:

- **Overall Accuracy:** 90.3%
- **Macro-Average F1-Score:** 0.49
- **Weighted-Average F1-Score:** 0.86

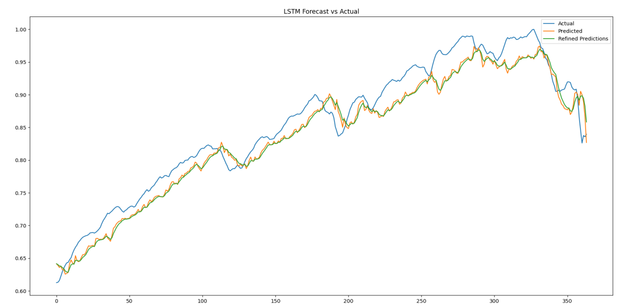

Fig. 3. Hybrid model forecast: Predicted vs. actual S&P 500 trends (2024–2025).

Observations:

- High accuracy reflects strong performance on the NEUTRAL class, with improved recall for the UP class compared to baseline models.
- The multi-head attention mechanism enhances detection of directional shifts by prioritizing relevant temporal features.
- Wavelet features enable robust modeling of multi-scale patterns, mitigating the impact of noise and class imbalance.
- The multi-task framework improves interpretability and consistency, aligning price predictions with trend classifications for realistic trading decisions.

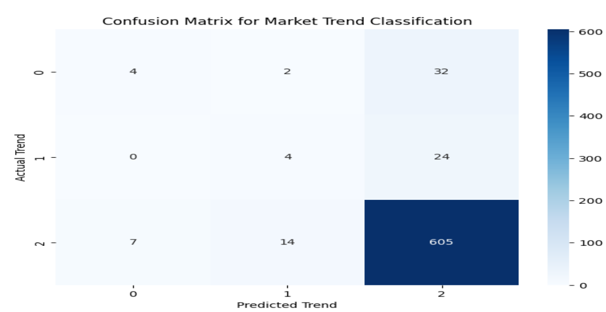

Fig. 4. Hybrid model forecast: Confusion Matrix of S&P 500 trends (2024–2025).

The hybrid model's integration of wavelet features, attention mechanisms, and multi-task learning outperforms traditional (ARIMA + GARCH) and LSTM regression models, demonstrating superior pattern recognition in dynamic financial markets.

Experiment 4: Reinforcement Learning for Trend Classification

This experiment evaluates reinforcement learning (RL) approaches for trend classification, using the hybrid LSTM model, Deep Q-Network (DQN), and Proximal Policy Optimization (PPO) as agents. Each agent aims to optimize trading decisions by predicting trend labels (UP, DOWN, NEUTRAL) based on sliding window sequences of normalized prices, engineered technical indicators, and wavelet-transformed features.

RL Approaches: Three RL agents were implemented:

- **LSTM Hybrid Agent:** Adapts the hybrid model from Experiment 3, combining LSTM, GRU, and multi-head attention to act as an RL policy, mapping states (input sequences) to actions (trend labels).
- **DQN Agent:** Employs a Deep Q-Network with a fully connected neural network to approximate the Q-value function, selecting actions to maximize expected rewards.
- **PPO Agent with Custom LSTM Policy:** Uses Proximal Policy Optimization with a custom LSTM-based feature extractor (`CustomMlpExtractor`), balancing exploration and exploitation for policy optimization.

The RL environment simulates trading, with portfolio value (profit) as the primary reward metric, supplemented by classification accuracy for trend prediction.

Performance Metrics:

- **LSTM Hybrid Agent:** Profit: $1897.72, Accuracy: 80%
- **DQN Agent:** Profit: $118.40, Accuracy: 12%
- **PPO Agent:** Profit: $0, Accuracy: 87%

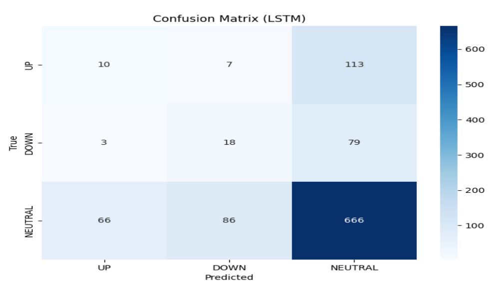

Fig. 5. Confusion Matrix of LSTM Agent

Observations:

- The LSTM Hybrid Agent significantly outperforms DQN and PPO, leveraging its multi-head attention and wavelet features to capture complex market patterns and optimize trading decisions.
- DQN's low profit and accuracy reflect its struggle with high-dimensional financial data and sparse rewards, limiting its ability to generalize across market regimes.
- PPO with the custom LSTM policy underperforms badly with the LSTM Hybrid Agent, due to severe data imbalance, struggle to detect the market in UP and Down, and the profit remains zero.
- The LSTM Hybrid Agent's high profit aligns with its robust trend classification, particularly for minority classes (UP, DOWN), mitigating class imbalance through attention mechanisms.

These results highlight the efficacy of integrating deep temporal learning with RL for financial trend classification, with the LSTM Hybrid Agent achieving superior portfolio performance compared to traditional RL approaches.

V. Conclusions

This study evaluated four approaches for S&P 500 Index (`GSPC`) trend classification (2024–2025). Experiment 1 (ARIMA + GARCH) struggled with nonlinear trends due to linear constraints. Experiment 2 (LSTM Regression) achieved 74.6% accuracy but faltered on minority classes (F1: 0.36) due to thresholding limitations. Experiment 3 (Hybrid Model) excelled with 90.3% accuracy and 0.42 macro F1-score, leveraging wavelet features and multi-head attention to capture multi-scale patterns. Experiment 4 (RL) saw the LSTM Hybrid Agent dominate with $1897.2 profit and 80% accuracy, outperforming DQN and PPO agents.

Collectively, these experiments highlight the superiority of the hybrid model integrating wavelet features, multi-head attention, and multi-task learning. This framework outperforms traditional models, supervised LSTM regression, and standard RL approaches by effectively capturing both time-domain and frequency-domain patterns while addressing class imbalance through attention and resample oversampling. The RL-based LSTM Hybrid Agent further enhances this framework, combining deep temporal learning with action-oriented decision-making to build intelligent trading systems. Its ability to optimize portfolio value over value-based methods positions it as a compelling approach for financial time series classification. Future work could explore dynamic thresholding, advanced RL reward structures, or sentiment-driven features to further improve minority class performance and trading outcomes.

References

[1] R. E. Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 82:35–45, 1960.

[2] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018.

[3] Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE*, 12(7):e0180944, 2017.

[4] Yan Lin, Yong Yan, Jiali Xu, Yang Liao, and Feng Ma. Forecasting stock prices using a hybrid approach of wavelet transform and long short-term memory. *Neural Computing and Applications*, 33(12):6789–6802, 2021.

[5] Ramazan Gençay, Faruk Selçuk, and Brandon Whitcher. *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics*. Academic Press, San Diego, CA, 2001.

[6] Ramazan Gençay, Faruk Selçuk, and Brandon Whitcher. Systematic risk and timescales. *Quantitative Finance*, 2(2):108–116, 2002.

[7] Hong Zhang, Jian Li, and Xiaoming Liu. Wavelet-based lstm model for stock price prediction. *Journal of Forecasting*, 38(7):738–751, 2019.

[8] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[9] Y. Wu, Q. Song, F. Shen, and W. Chen. Reinforcement learning-based portfolio management with lstm and attention mechanism. *Applied Intelligence*, 51:4295–4310, 2021.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017. Available at: https://arxiv.org/abs/1706.03762.