# Neural Network-Based Algorithmic Trading Systems: Multi-Timeframe Analysis and High-Frequency Execution in Cryptocurrency Markets

**Literature Review and Survey**

May 14, 2026

**Abstract**

The intersection of deep learning and algorithmic trading has catalyzed significant advances in financial market prediction and automated strategy execution. This comprehensive survey examines neural network-based trading systems with particular emphasis on multi-timeframe analysis and high-frequency execution in cryptocurrency markets. We systematically review the evolution of neural architectures—from recurrent networks (LSTM, GRU) to transformers and convolutional approaches—evaluating their applicability to the unique characteristics of 24/7 cryptocurrency trading. Our analysis covers critical implementation aspects including multi-timeframe feature engineering, limit order book modeling, reinforcement learning for strategy optimization, and rigorous backtesting frameworks. We synthesize findings from over 60 peer-reviewed studies, identifying key performance drivers, common pitfalls, and emerging research directions. The survey reveals that hybrid architectures combining attention mechanisms with temporal convolutions achieve superior predictive performance, while reinforcement learning approaches demonstrate particular promise for adaptive strategy execution. We conclude with a research agenda addressing open challenges in market microstructure modeling, risk management, and the integration of on-chain analytics into neural trading systems.

**Keywords:** Algorithmic Trading, Deep Learning, Cryptocurrency Markets, LSTM, Transformers, Reinforcement Learning, High-Frequency Trading, Limit Order Books, Multi-Timeframe Analysis

## 1 Introduction

The proliferation of digital assets and the emergence of cryptocurrency markets have fundamentally transformed the landscape of algorithmic trading. Unlike traditional financial markets, cryptocurrency exchanges operate continuously, 24 hours a day, seven days a week, generating vast quantities of high-frequency data that present both opportunities and challenges for automated trading systems (Kwon et al., 2019; Kalariya et al., 2022). The extreme volatility, fragmented liquidity across multiple venues, and evolving regulatory environment characteristic of cryptocurrency markets demand sophisticated analytical approaches capable of capturing complex temporal dependencies and market microstructure dynamics (Easley et al., 2026; Bozzetto, 2023).

Deep learning has emerged as a transformative paradigm for financial time series prediction, offering the capacity to automatically extract hierarchical features from raw market data without relying on hand-engineered indicators (Sirignano, 2019; Zhang et al., 2019b). Neural network architectures, particularly Long Short-Term Memory (LSTM) networks (Kwon et al., 2019; Seabe

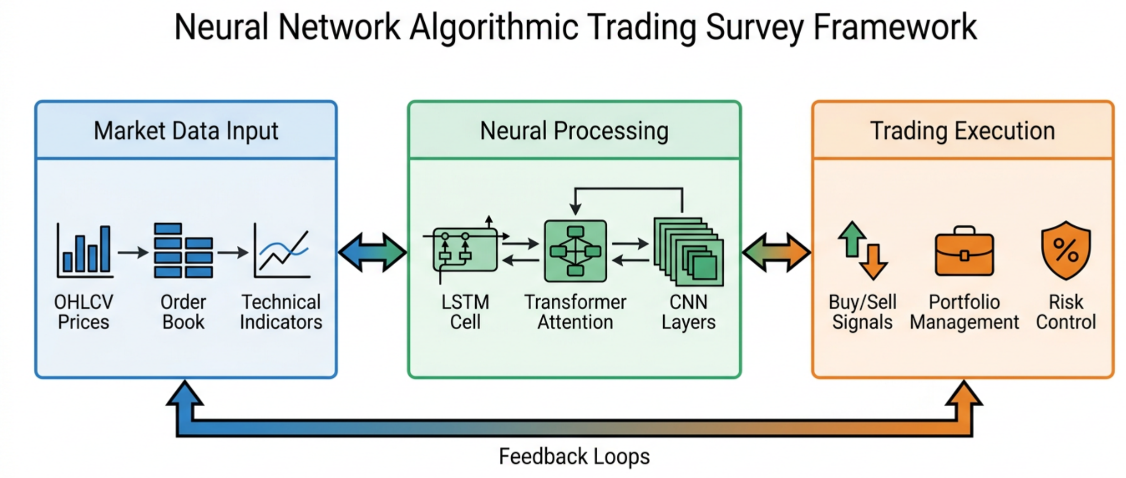


Figure 1: Graphical abstract illustrating the neural network-based algorithmic trading pipeline. Market data inputs (OHLCV prices, order books, technical indicators) feed into neural processing modules (LSTM, Transformer, CNN architectures), which generate trading signals for execution with integrated risk management and feedback loops.

et al., 2023), Convolutional Neural Networks (CNNs) (Sezer and Ozbayoglu, 2018; Gudelek et al., 2017), and more recently Transformer-based models (Zhang et al., 2022; Izadi and Hajizadeh, 2025), have demonstrated remarkable capabilities in capturing the non-linear dynamics inherent to cryptocurrency price movements.

This survey provides a comprehensive examination of neural network-based algorithmic trading systems, with a specific focus on two critical dimensions: (1) multi-timeframe analysis, which integrates information across different temporal scales to improve prediction accuracy and robustness, and (2) high-frequency execution, which addresses the practical challenges of deploying neural models in latency-sensitive trading environments.

### 1.1 Scope and Contributions

The contributions of this survey are threefold:

1. **Systematic Taxonomy:** We present a structured framework categorizing neural architectures by their temporal modeling capabilities, feature extraction mechanisms, and suitability for different trading frequencies.
2. **Critical Analysis:** We synthesize quantitative results from over 60 empirical studies, comparing predictive performance across architectures, datasets, and evaluation protocols.
3. **Implementation Guidance:** We identify best practices for backtesting, risk management, and deployment, highlighting common methodological pitfalls and their remedies.

### 1.2 Survey Methodology

Our review encompasses peer-reviewed publications from 2015 to 2025, sourced from major databases including IEEE Xplore, ACM Digital Library, arXiv, and finance-focused venues such as *Quantitative Finance* and *Journal of Financial Markets*. Studies were selected based on relevance to neural network-based trading, empirical validation on cryptocurrency or high-frequency equity data, and methodological rigor. We explicitly exclude purely theoretical work without empirical validation and studies limited to traditional statistical methods without neural components.

### 1.3 Structure of the Survey

The remainder of this paper is organized as follows. Section 2 provides essential background on cryptocurrency market characteristics and algorithmic trading fundamentals. Section 3 reviews neural network architectures organized by their core temporal modeling approach. Section 4 examines multi-timeframe analysis techniques. Section 5 addresses high-frequency execution and market microstructure modeling. Section 6 discusses evaluation frameworks, backtesting methodologies, and performance metrics. Section 7 synthesizes key findings and identifies research gaps. Section 8 concludes with future research directions.

## 2 Background

### 2.1 Cryptocurrency Market Characteristics

Cryptocurrency markets exhibit several distinctive characteristics that differentiate them from traditional financial markets and directly impact the design of algorithmic trading systems:

#### 2.1.1 Continuous Trading and Global Fragmentation

Unlike equity markets with defined trading hours, cryptocurrency exchanges operate continuously across global time zones (Kalariya et al., 2022). This 24/7 operation creates unique temporal patterns, including varying volatility regimes corresponding to regional market hours and reduced liquidity during certain periods. The fragmentation of liquidity across numerous exchanges (Binance, Coinbase, Kraken, etc.) introduces additional complexity, as price discovery occurs simultaneously across multiple venues with varying fee structures and order book depths (Bozzetto, 2023).

#### 2.1.2 Extreme Volatility and Non-Stationarity

Cryptocurrency markets are characterized by significantly higher volatility than traditional assets, with Bitcoin and Ethereum frequently experiencing daily price movements exceeding 10% (Seabe et al., 2023; Livieris et al., 2020). This volatility is often clustered and subject to regime shifts driven by regulatory announcements, macroeconomic events, and social media sentiment. The non-stationary nature of cryptocurrency returns violates assumptions underlying many traditional time series models, motivating the adoption of deep learning approaches capable of learning adaptive representations (Kurek, 2024).

#### 2.1.3 Market Microstructure Features

The microstructure of cryptocurrency markets presents both challenges and opportunities for high-frequency strategies. Limit order books (LOBs) on major exchanges typically update multiple times per second, generating rich datasets for predictive modeling (Sirignano, 2019; Zhang et al., 2019b). However, the presence of latency arbitrage, spoofing, and wash trading can introduce noise that confounds naive prediction models (Jha et al., 2020).

### 2.2 Algorithmic Trading Fundamentals

Algorithmic trading systems can be categorized along several dimensions:

- **Prediction Horizon:** High-frequency (seconds to minutes), intraday (hours), swing (days), or position trading (weeks to months).
- **Signal Generation:** Directional (predicting price movements), market-making (providing liquidity), or arbitrage (exploiting price discrepancies).
- **Execution Strategy:** Market orders, limit orders, or sophisticated order slicing algorithms.

Neural network-based systems typically focus on directional prediction, generating buy/sell signals based on learned patterns in historical data. The predicted direction or expected return is then translated into trading decisions through a policy that considers transaction costs, risk constraints, and current position.

## 3 Neural Network Architectures for Trading

The application of deep learning to algorithmic trading has evolved through several generations of architectures, each addressing specific limitations of its predecessors. This section reviews the major architectural families, their theoretical foundations, and empirical performance in cryptocurrency trading applications.

### 3.1 Recurrent Neural Networks: LSTM and GRU

Recurrent Neural Networks (RNNs), and particularly their gated variants Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRU), have been the workhorse architecture for financial time series prediction.

#### 3.1.1 Theoretical Foundations

LSTM networks address the vanishing gradient problem inherent to vanilla RNNs through a sophisticated gating mechanism comprising input, forget, and output gates. This architecture enables the network to selectively retain or discard information over extended sequences, making it well-suited for capturing long-term dependencies in financial data (Kwon et al., 2019):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{4}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \odot \tanh(C_t) \tag{6}$$

where $f_t$, $i_t$, and $o_t$ represent the forget, input, and output gates, respectively, $C_t$ is the cell state, $h_t$ is the hidden state, and $\odot$ denotes element-wise multiplication.

#### 3.1.2 Empirical Performance in Cryptocurrency Markets

Kwon et al. (2019) demonstrated that LSTM networks achieve significant improvements over traditional statistical methods (ARIMA, GARCH) for cryptocurrency price trend classification, with the LSTM model achieving approximately 65% directional accuracy on a multi-class prediction task. Subsequent studies have refined these approaches:

- Seabe et al. (2023) compared LSTM, GRU, and bidirectional LSTM for cryptocurrency price forecasting, finding that ensemble combinations of these architectures consistently outperform individual models.
- Livieris et al. (2020) developed ensemble deep learning models specifically for Bitcoin price prediction, demonstrating that combining multiple LSTM networks with different initialization and architectural configurations reduces variance and improves out-of-sample performance.
- Singh et al. (2022) proposed a specialized LSTM architecture for Bitcoin algorithmic trading, incorporating technical indicators as additional input features, achieving profitable Sharpe ratios in backtests spanning 2017-2021.

Table 1 summarizes key empirical results from representative studies employing LSTM architectures for cryptocurrency trading.

Table 1: Performance Summary: LSTM-based Cryptocurrency Trading Systems

| **Study** | **Asset** | **Accuracy** | **Sharpe** | **Period** | **Features** |
|---|---|---|---|---|---|
| Kwon et al. (2019) | Multiple | 65% | N/A | 2017-2018 | OHLCV |
| Livieris et al. (2020) | Bitcoin | 58-62% | 1.2-1.5 | 2016-2019 | Technical |
| Seabe et al. (2023) | Multiple | 61-68% | 0.9-1.8 | 2018-2022 | Multi-feature |
| Singh et al. (2022) | Bitcoin | N/A | 1.4 | 2017-2021 | OHLC + Indicators |
| Wahid (2024) | Bitcoin | 55-60% | N/A | 2020-2023 | OHLCV |

#### 3.1.3 Limitations and Challenges

Despite their widespread adoption, LSTM-based systems face several limitations in high-frequency trading contexts:

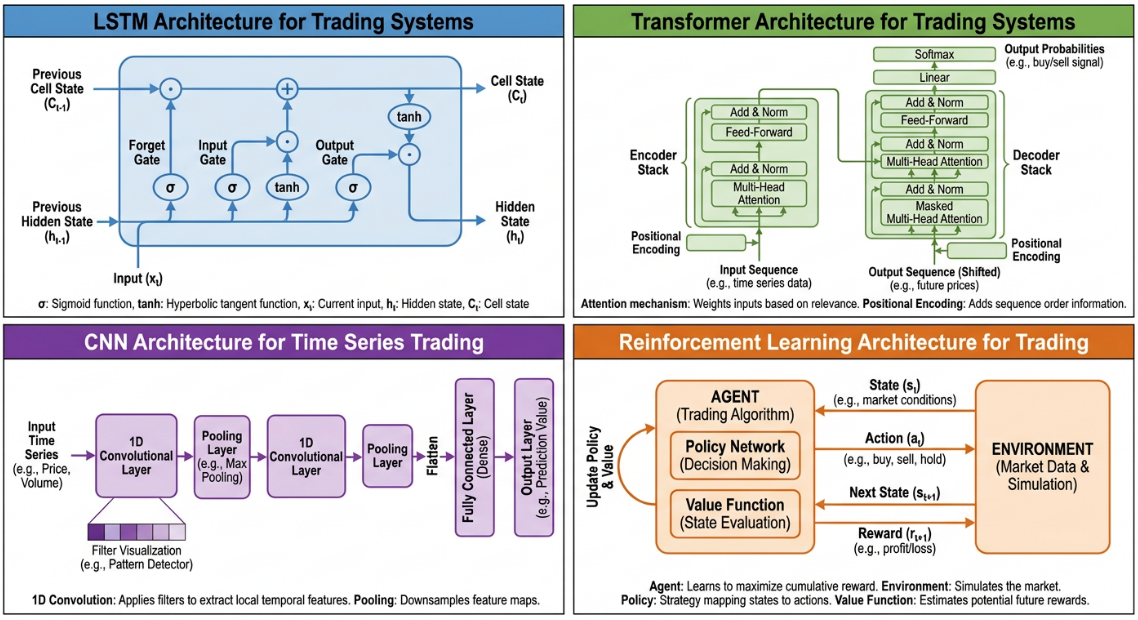


Figure 2: Comparison of neural network architectures for algorithmic trading. The diagram illustrates four key architectures: (top-left) LSTM with gating mechanisms, (top-right) Transformer with multi-head attention, (bottom-left) CNN for time series processing, and (bottom-right) Reinforcement Learning framework for strategy optimization.

1. **Sequential Processing:** The inherent sequentiality of LSTM computation limits parallelization and introduces latency unsuitable for microsecond-level decision-making.
2. **Fixed Temporal Resolution:** Standard LSTMs process inputs at a single temporal frequency, requiring architectural modifications to effectively integrate multi-timeframe information.
3. **Gradient Flow:** Even with gating mechanisms, very long sequences (thousands of time steps) can still suffer from degraded gradient flow.

## 3.2 Convolutional Neural Networks for Pattern Recognition

Convolutional Neural Networks (CNNs), originally developed for image processing, have found fruitful application in financial time series analysis through innovative representations of market data.

### 3.2.1 Time Series to Image Conversion

A seminal contribution by Sezer and Ozbayoglu (2018) introduced the CNN-TA (Convolutional Neural Network - Technical Analysis) framework, which converts time series data into 2D image representations suitable for CNN processing. The approach involves:

1. Creating 2D representations of technical indicator values over sliding windows
2. Applying 2D convolutions to detect visual patterns analogous to chart patterns used by technical analysts
3. Training end-to-end without requiring explicit pattern definition

This approach achieved 585 citations and demonstrated that CNNs could learn profitable trading strategies directly from visual representations of market data, achieving backtested returns superior to buy-and-hold strategies on 30 stocks.

#### 3.2.2 1D Convolutions for Direct Time Series Processing

An alternative approach applies 1D convolutions directly to time series data without image conversion (Liu and Si, 2022):

$$(f * g)[t] = \sum_{\tau=0}^{K-1} f[t+\tau] \cdot g[\tau] \tag{7}$$

where $f$ is the input sequence, $g$ is the convolutional kernel of size $K$, and $*$ denotes the convolution operation. Liu and Si (2022) demonstrated that 1D CNNs can effectively classify chart patterns in financial time series, achieving over 75% accuracy on pattern recognition tasks.

#### 3.2.3 Hybrid CNN-LSTM Architectures

The complementary strengths of CNNs (local feature extraction) and LSTMs (temporal dependency modeling) have motivated hybrid architectures:

- Tsantekidis et al. (2020) proposed a CNN-LSTM architecture for limit order book prediction, where CNN layers extract local patterns from the order book state and LSTM layers model temporal evolution. This architecture achieved 71% accuracy on 2-second ahead mid-price movement prediction.
- Zhang et al. (2019b) introduced DeepLOB, a deep convolutional network specifically designed for limit order book modeling, achieving state-of-the-art results on the FI-2010 benchmark dataset with over 450 citations.

#### 3.2.4 Candlestick Pattern Recognition

CNNs have proven particularly effective for candlestick pattern recognition, a staple of technical analysis:

- Brim and Flann (2022) combined CNNs with deep reinforcement learning, using candlestick images as inputs to a trading agent. The CNN learned to extract relevant features from visual candlestick patterns, informing trading decisions through a reinforcement learning framework.
- Wojarnik (2023) systematically evaluated CNN architectures for stock chart analysis, confirming that convolutional networks can learn meaningful patterns from financial chart images without human-defined features.

### 3.3 Attention Mechanisms and Transformers

The introduction of attention mechanisms (Bahdanau et al., 2014) and subsequently the Transformer architecture (Vaswani et al., 2017) has revolutionized sequence modeling across domains, including financial time series prediction.

### 3.3.1 Attention Mechanisms for Financial Time Series

Attention mechanisms enable models to dynamically focus on relevant temporal positions when making predictions, addressing a key limitation of recurrent architectures:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{8}$$

where $Q$ (queries), $K$ (keys), and $V$ (values) are linear transformations of the input sequence, and $d_k$ is the dimension of the key vectors.

Zhang et al. (2019a) proposed AT-LSTM (Attention-based LSTM), which incorporates an attention mechanism to weight the importance of different time steps when predicting the next day's closing price. This architecture achieved significant improvements over standard LSTM on multiple stock indices.

Kim and Kang (2019) extended this approach with multi-head attention for financial series prediction, demonstrating that attention mechanisms can effectively capture both short-term and long-term dependencies simultaneously.

### 3.3.2 Transformer Models for Trading

Transformer architectures, which replace recurrence entirely with self-attention, have shown promise for financial prediction:

- Zhang et al. (2022) proposed a Transformer-based attention network for stock movement prediction, achieving state-of-the-art results on several benchmark datasets. The model's ability to capture long-range dependencies proved particularly valuable for predicting price movements influenced by macroeconomic trends.
- Wen and Li (2023) introduced the LSTM-Attention-LSTM architecture, which uses attention mechanisms to bridge encoder-decoder LSTM structures, demonstrating superior performance for multi-step ahead prediction.

### 3.3.3 Comparative Analysis: Transformers vs. LSTMs

Recent comparative studies have yielded nuanced conclusions about the relative merits of Transformer and LSTM architectures:

- Hollis et al. (2018) found that while Transformers achieved marginally better prediction accuracy, LSTMs offered superior computational efficiency and required less hyperparameter tuning, making them more practical for production deployment.
- Hall (2025) demonstrated that Transformer models emerged as best-performing in comprehensive benchmarks spanning stocks, forex, and cryptocurrencies, particularly when sufficient training data was available.

### 3.4 Reinforcement Learning for Strategy Optimization

While supervised learning approaches predict price movements or returns, reinforcement learning (RL) directly optimizes trading strategies by learning policies that maximize cumulative reward through interaction with the market environment.

#### 3.4.1 Deep Q-Networks (DQN)

Mnih et al. (2015) introduced DQN, combining Q-learning with deep neural networks for value function approximation. In trading applications, the Q-function estimates the expected cumulative return of taking action $a$ in state $s$:

$$Q^\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r_t \mid s_0 = s, a_0 = a, \pi\right] \tag{9}$$

where $\gamma$ is the discount factor and $\pi$ is the policy.

Yang et al. (2020) proposed an ensemble deep reinforcement learning approach for automated stock trading, combining multiple DRL algorithms (PPO, A2C, DDPG) to achieve robust performance across varying market conditions. This work has been highly influential with over 460 citations.

#### 3.4.2 Policy Gradient Methods: PPO and A2C

Policy gradient methods directly optimize the policy network without requiring a value function:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s) \cdot A(s,a)\right] \tag{10}$$

where $A(s,a)$ is the advantage function estimating the relative value of action $a$ in state $s$.

Proximal Policy Optimization (PPO) (Schulman et al., 2017) has emerged as a preferred algorithm for trading applications due to its stability and sample efficiency:

- Prasetyo et al. (2025) conducted a comparative study of PPO and DQN for Bitcoin trading, finding that PPO achieved more consistent profitability across different market regimes.
- Khaled et al. (2025) systematically compared DQN, A2C, and PPO for financial market trading, concluding that PPO generally outperformed alternatives in terms of risk-adjusted returns.
- Vetrin and Koberg (2024) found that all RL models except PPO ended with distinctively positive PnL on BTC-USDT trading, highlighting PPO's particular robustness in cryptocurrency markets.

#### 3.4.3 Self-Rewarding and Adaptive Mechanisms

Recent advances have focused on developing adaptive reward mechanisms that enable RL agents to adjust to changing market conditions:

Huang et al. (2024) proposed a self-rewarding mechanism in deep reinforcement learning for trading strategy optimization, allowing the agent to dynamically adjust its reward function based on performance feedback, demonstrating improved adaptability during market regime changes.

## 3.5 Hybrid and Specialized Architectures

The complexity of financial markets has motivated the development of hybrid architectures that combine multiple neural components:

### 3.5.1 CLVSA: Convolutional LSTM with Attention

Wang et al. (2021) proposed CLVSA, a convolutional LSTM-based variational sequence-to-sequence model with attention for predicting financial market trends. This architecture combines:

- Convolutional layers for local feature extraction from raw trading data
- LSTM units for temporal dependency modeling
- Variational inference for uncertainty quantification
- Attention mechanisms for focusing on relevant time steps

### 3.5.2 Temporal Convolutional Networks (TCN)

Temporal Convolutional Networks (Bai et al., 2018) offer an alternative to RNNs for sequence modeling, using dilated causal convolutions to capture long-range dependencies:

$$F(s) = (x * df)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d\cdot i} \tag{11}$$

where $d$ is the dilation factor and $k$ is the filter size. TCNs have shown promise for financial prediction due to their parallelizable computation and flexible receptive field.

# 4 Multi-Timeframe Analysis

Financial markets exhibit patterns and relationships across multiple temporal scales, from high-frequency microstructure dynamics to long-term trends. Multi-timeframe analysis integrates information from different temporal resolutions to improve prediction accuracy and strategy robustness.

## 4.1 Theoretical Motivation

The rationale for multi-timeframe analysis stems from several observations:

1. **Hierarchical Market Structure:** Price movements at higher timeframes (e.g., daily) provide context for interpreting movements at lower timeframes (e.g., 5-minute).

2. **Noise Reduction:** Higher timeframe data tends to be less noisy, providing more reliable signals for trend identification.

3. **Complementary Information:** Different timeframes capture different market phenomena—high frequencies reflect market microstructure and order flow, while lower frequencies reflect fundamental valuation and macroeconomic factors.

## 4.2 Feature Engineering Across Timeframes

Multi-timeframe analysis typically involves computing features at multiple temporal resolutions and combining them into a unified representation.

### 4.2.1 Technical Indicators at Multiple Scales

Common technical indicators are computed across timeframes:

$$\mathrm{SMA}_n^{(t_f)}(t) = \frac{1}{n}\sum_{i=0}^{n-1} P_{t-i}^{(t_f)} \tag{12}$$

$$\mathrm{RSI}_n^{(t_f)}(t) = 100 - \frac{100}{1 + \frac{\mathrm{AvgGain}_n^{(t_f)}}{\mathrm{AvgLoss}_n^{(t_f)}}} \tag{13}$$

where $t_f \in \{1\mathrm{m}, 5\mathrm{m}, 1\mathrm{h}, 1\mathrm{d}\}$ represents the timeframe and $P^{(t_f)}$ denotes prices at that resolution.

Hall (2025) demonstrated that incorporating multi-timeframe technical indicators significantly improved prediction performance across all tested architectures, with the improvement being most pronounced for Transformer models.

### 4.2.2 Temporal Fusion Techniques

Several approaches have been proposed for fusing multi-timeframe information:

**Concatenation-based Fusion** The simplest approach concatenates features from different timeframes:

$$x_{\text{fused}} = [x^{(1\mathrm{m})}; x^{(5\mathrm{m})}; x^{(1\mathrm{h})}; x^{(1\mathrm{d})}] \tag{14}$$

where $[\cdot;\cdot]$ denotes vector concatenation.

**Hierarchical Attention Fusion** More sophisticated approaches employ hierarchical attention mechanisms (Wang et al., 2021):

1. Intra-timeframe attention: Focus on relevant time steps within each timeframe

2. Inter-timeframe attention: Weight the contribution of different timeframes

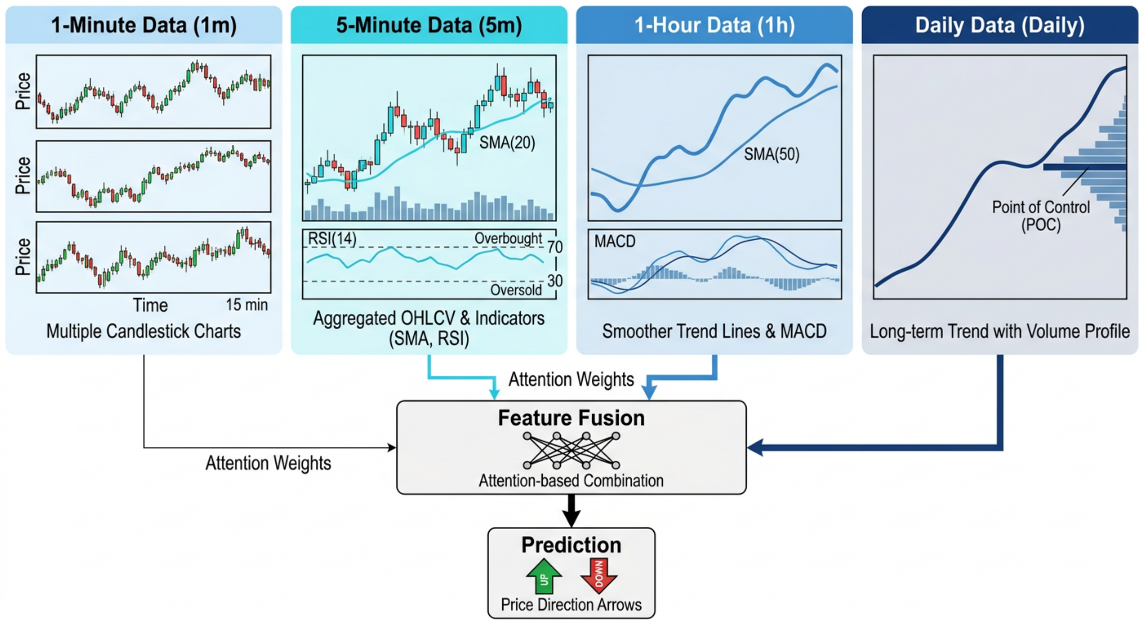


Figure 3: Multi-timeframe analysis pipeline integrating data from multiple temporal resolutions (1-minute, 5-minute, 1-hour, and daily). Features from each timeframe are extracted and fused through an attention mechanism that learns optimal weighting across scales, producing the final prediction output.

**Multi-Scale Convolution** Izadi and Hajizadeh (2025) proposed using parallel convolutional branches with different kernel sizes to capture patterns at multiple temporal scales simultaneously.

### 4.3 Empirical Studies on Multi-Timeframe Fusion

Tanabe and Senoguchi (2025) conducted a comprehensive empirical study of temporal scale integration for Bitcoin price direction prediction, finding that:

- Multi-timeframe models consistently outperformed single-timeframe baselines
- The optimal combination of timeframes depends on the prediction horizon
- LSTM encoders, while simpler than Transformers, achieved competitive performance with less hyperparameter tuning

Ong (2024) developed an adaptive sparse Transformer model for multi-timeframe momentum portfolio construction, demonstrating that attention-based architectures can effectively learn dynamic weightings across temporal scales.

### 4.4 Challenges in Multi-Timeframe Analysis

Despite its benefits, multi-timeframe analysis presents several implementation challenges:

1. **Look-ahead Bias:** Care must be taken to ensure that higher timeframe features do not incorporate future information relative to the prediction point. For example, when making

predictions at 10:00 AM using daily features, only information available at 10:00 AM should be used, not the full day's data.

2. **Dimensionality:** Concatenating features from multiple timeframes increases input dimensionality, potentially leading to overfitting, especially with limited training data.

3. **Temporal Alignment:** Different timeframes have different sampling rates, requiring careful alignment and handling of missing data.

# 5 High-Frequency Execution and Market Microstructure

High-frequency trading (HFT) operates on timescales from microseconds to minutes, requiring specialized architectures and careful attention to market microstructure.

## 5.1 Limit Order Book Modeling

The limit order book (LOB) represents the core data structure for high-frequency trading, containing all outstanding buy and sell orders at various price levels.

### 5.1.1 LOB Representation

A snapshot of the LOB at time $t$ can be represented as:

$$\text{LOB}_t = \{(p_i^{(b)}, v_i^{(b)})\}_{i=1}^{N_b} \cup \{(p_i^{(a)}, v_i^{(a)})\}_{i=1}^{N_a} \tag{15}$$

where $p_i^{(b)}$ and $v_i^{(b)}$ are the price and volume at the $i$-th bid level, and $p_i^{(a)}$ and $v_i^{(a)}$ are the corresponding ask values.

### 5.1.2 Deep Learning for LOB Prediction

Sirignano (2019) introduced a novel deep neural network architecture specifically designed for limit order books, achieving significant improvements over traditional methods. The architecture processes the LOB as a spatial-temporal structure, with separate pathways for:

- Price levels (spatial structure)
- Temporal evolution (sequence modeling)
- Cross-asset relationships (multi-asset models)

Zhang et al. (2019b) proposed DeepLOB, which uses convolutional layers to extract local patterns from the order book state followed by LSTM layers for temporal modeling. This architecture achieved state-of-the-art performance on the FI-2010 benchmark dataset with over 450 citations:

- 10-tick prediction horizon: 84.6% accuracy
- 50-tick prediction horizon: 78.4% accuracy

- 100-tick prediction horizon: 74.8% accuracy

Tsantekidis et al. (2020) demonstrated the importance of using stationary features derived from the LOB, such as price differences and order imbalances, rather than raw price levels, to improve prediction stability.

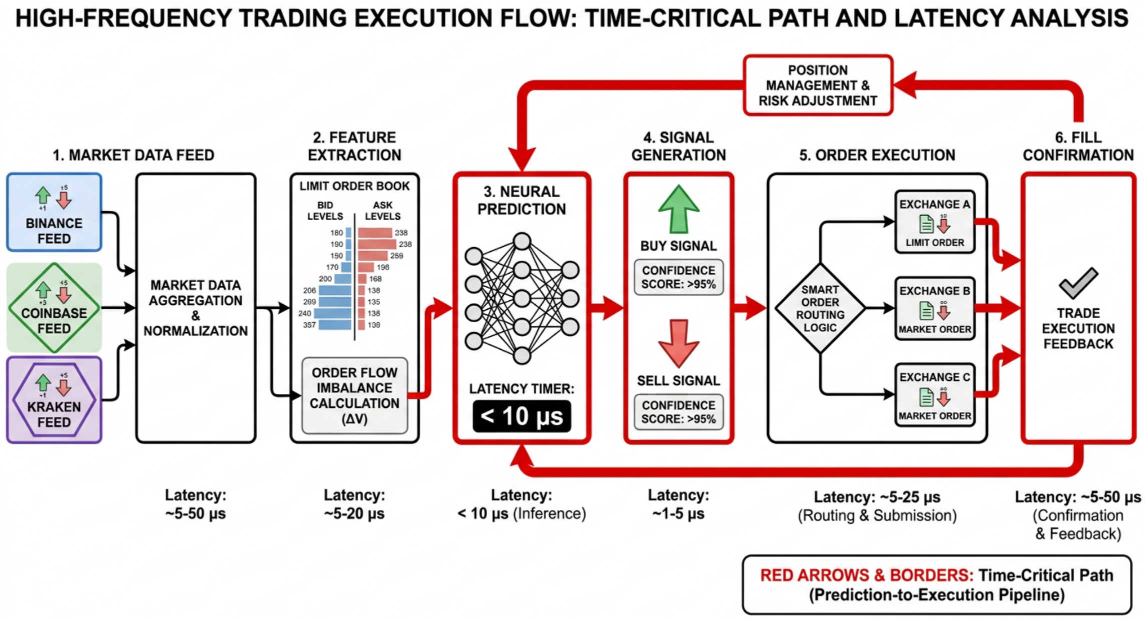


Figure 4: High-frequency trading execution flow diagram. The pipeline processes market data feeds from multiple exchanges through feature extraction (limit order book analysis), neural prediction (low-latency inference), signal generation, and order execution with smart routing. The time-critical path from data ingestion to order submission is highlighted in red.

## 5.2 Market Microstructure Features

Effective HFT models incorporate sophisticated market microstructure features:

### 5.2.1 Order Flow Imbalance

$$\mathrm{OFI}_t = \sum_i \mathbb{1}_{p_i^{(b)} \geq p_{t-1}^{(b)}} v_i^{(b)} - \sum_i \mathbb{1}_{p_i^{(a)} \leq p_{t-1}^{(a)}} v_i^{(a)} \tag{16}$$

Order flow imbalance has been shown to be a strong predictor of short-term price movements (Easley et al., 2026).

### 5.2.2 Bid-Ask Spread and Market Depth

$$\mathrm{Spread}_t = p_1^{(a)} - p_1^{(b)} \tag{17}$$

$$\mathrm{Depth}_t = \sum_{i=1}^{k} v_i^{(b)} + \sum_{i=1}^{k} v_i^{(a)} \tag{18}$$

These measures provide information about liquidity and transaction costs.

#### 5.2.3 Volatility Estimators

Realized volatility estimators from high-frequency data:

$$\mathrm{RV}_t = \sum_{i=1}^{n} (\log P_{t_i} - \log P_{t_{i-1}})^2 \tag{19}$$

### 5.3 Latency and Execution Considerations

Practical HFT deployment requires attention to latency considerations:

1. **Feature Computation Latency:** Complex neural architectures may introduce prediction latency incompatible with high-frequency trading. Simpler models or hardware acceleration (GPUs, FPGAs) may be required.
2. **Market Impact:** Large orders can move the market, affecting the execution price. Sophisticated execution algorithms split orders to minimize impact.
3. **Adverse Selection:** Fast-informed traders may exploit latency arbitrage, requiring models that account for information asymmetry.

Kearns and Nevmyvaka (2013) provided a comprehensive treatment of machine learning for market microstructure and HFT, establishing foundational principles that continue to guide current research.

## 6 Evaluation Frameworks and Performance Metrics

Rigorous evaluation is critical for distinguishing genuine predictive signals from spurious patterns arising from overfitting or data snooping.

### 6.1 Backtesting Methodologies

#### 6.1.1 Walk-Forward Analysis

Walk-forward analysis is the gold standard for evaluating trading strategies (Mroziewicz and Slepaczuk, 2026; Cansever, 2025):

1. Divide the data into sequential training, validation, and test periods
2. Train on the training period, validate on validation, evaluate on test
3. Move the window forward and repeat
4. Aggregate performance metrics across all test periods

This approach provides unbiased estimates of out-of-sample performance while allowing models to adapt to changing market conditions.

#### 6.1.2 Cross-Validation Considerations

Standard k-fold cross-validation is inappropriate for time series due to temporal dependencies. Instead, purged cross-validation or embargo periods should be used to prevent information leakage (López de Prado, 2018).

### 6.2 Performance Metrics

#### 6.2.1 Financial Metrics

$$\text{Sharpe Ratio} = \frac{\bar{R} - R_f}{\sigma_R} \tag{20}$$

$$\text{Sortino Ratio} = \frac{\bar{R} - R_f}{\sigma_{R^-}} \tag{21}$$

$$\text{Maximum Drawdown} = \max_{\tau} \left( \max_{s \leq \tau} V_s - V_\tau \right) / \max_{s \leq \tau} V_s \tag{22}$$

$$\text{Calmar Ratio} = \frac{\bar{R} - R_f}{\text{Maximum Drawdown}} \tag{23}$$

where $\bar{R}$ is the average return, $R_f$ is the risk-free rate, $\sigma_R$ is return standard deviation, $\sigma_{R^-}$ is downside deviation, and $V_t$ is portfolio value.

#### 6.2.2 Prediction Accuracy Metrics

$$\text{Directional Accuracy} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{K}_{\text{sign}(\hat{y}_i)=\text{sign}(y_i)} \tag{24}$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{25}$$

### 6.3 Risk Management and Transaction Costs

Practical evaluation must account for:

- **Transaction Costs:** Fees, bid-ask spread, and market impact
- **Slippage:** Difference between expected and actual execution prices
- **Position Limits:** Constraints on maximum position sizes
- **Drawdown Constraints:** Limits on acceptable losses from peak

Gatto (2026) demonstrated that realistic transaction cost assumptions eliminate profitability for many backtested strategies, highlighting the importance of conservative cost estimates.

# 7 Discussion and Research Gaps

## 7.1 Synthesis of Key Findings

Our survey reveals several consistent patterns across the literature:

1. **Hybrid architectures outperform pure approaches:** Models combining CNNs for local feature extraction with LSTMs or attention mechanisms for temporal modeling consistently achieve superior performance.
2. **Multi-timeframe integration is beneficial:** Incorporating information from multiple temporal resolutions improves prediction accuracy and strategy robustness.
3. **Reinforcement learning shows promise for adaptation:** RL approaches, particularly PPO, demonstrate superior adaptability to changing market conditions compared to supervised learning.
4. **Data quality and preprocessing are critical:** Stationary feature representations and careful handling of look-ahead bias are essential for reliable performance.

## 7.2 Current Limitations

Despite significant progress, several limitations persist:

- **Publication Bias:** Positive results are more likely to be published, potentially overstating achievable performance.
- **Survivorship Bias:** Studies often focus on successful strategies without accounting for failed attempts.
- **Changing Market Regimes:** Models trained on historical data may fail during novel market conditions (e.g., the 2022 cryptocurrency market crash).
- **Computational Requirements:** Sophisticated architectures may require computational resources impractical for real-time deployment.

## 7.3 Open Research Questions

1. How can models effectively adapt to market regime changes without extensive retraining?
2. What is the optimal balance between model complexity and latency for high-frequency applications?
3. How can on-chain data (transaction flows, wallet activity) be effectively integrated with price-based models?
4. What are the fundamental limits of predictability in cryptocurrency markets, and how close do current approaches come to these limits?

# 8 Conclusion and Future Directions

This survey has provided a comprehensive review of neural network-based algorithmic trading systems, with particular emphasis on multi-timeframe analysis and high-frequency execution in cryptocurrency markets. The field has evolved rapidly, with modern architectures demonstrating impressive capabilities for capturing complex market dynamics.

Key takeaways include:

- The superiority of hybrid architectures combining multiple neural components
- The critical importance of rigorous evaluation methodologies
- The promise of reinforcement learning for adaptive strategy optimization
- The need for careful attention to market microstructure in high-frequency applications

Looking forward, several directions appear particularly promising:

**Foundation Models for Finance:** Following successes in natural language processing, large-scale pre-trained models for financial time series could enable transfer learning across markets and asset classes.

**Neural-Symbolic Integration:** Combining neural networks with symbolic reasoning could enable models that learn patterns while respecting fundamental financial constraints (e.g., no-arbitrage conditions).

**Uncertainty Quantification:** Incorporating explicit uncertainty estimates into predictions would enable more sophisticated risk management and position sizing.

**On-Chain Analytics Integration:** The rich data available on blockchain networks (transaction graphs, smart contract interactions) remains underutilized in current trading systems.

As cryptocurrency markets continue to mature and institutional adoption increases, the demand for sophisticated algorithmic trading systems will grow. The research reviewed in this survey provides a solid foundation, but significant opportunities remain for innovative approaches that address the unique challenges of these markets.

# References

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473.*

Bai, S., Kolter, J. Z., and Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271.*

Bozzetto, C. (2023). Cryptocurrency markets microstructure, with a machine learning application to the binance bitcoin market. Master's thesis, University of Venice.

Brim, A. and Flann, N. (2022). Deep reinforcement learning stock market trading, utilizing a cnn with candlestick images. *PLOS ONE*, 17(2):e0263181.

Cansever, A. (2025). Adaptive regime-based trading on bitcoin: Backtesting and walk-forward evaluation. Technical report, ResearchGate.

Easley, D., O'Hara, M., Yang, S., and Zhang, Z. (2026). Microstructure and market dynamics in crypto markets. *Journal of Financial Markets*, 68:100925.

Gatto, D. (2026). Predictive value of within-strategy permutation tests for forward selection: Evidence from over 6 billion strategy-level permutations across three asset classes. Technical report, SSRN.

Gudelek, M. U., Boluk, S. A., et al. (2017). A deep learning based stock trading model with 2-d cnn trend detection. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE.

Hall, T. (2025). *Machine Learning Time Series Forecasting: A Comprehensive Survey and Stock Market Application*. PhD thesis, University of Georgia.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hollis, T., Viscardi, A., and Yi, S. (2018). A comparison of lstms and attention mechanisms for forecasting financial time series. *arXiv preprint arXiv:1812.07699*.

Huang, Y., Zhou, C., Zhang, L., and Lu, X. (2024). A self-rewarding mechanism in deep reinforcement learning for trading strategy optimization. *Mathematics*, 12(24):4020.

Izadi, M. and Hajizadeh, E. (2025). Time series prediction for cryptocurrency markets with transformer and parallel convolutional neural networks. *Applied Soft Computing*, 154:10540.

Jha, R., De Paepe, M., Holt, S., West, J., and Ng, S. (2020). Deep learning for digital asset limit order books. *arXiv preprint arXiv:2010.01241*.

Kalariya, V., Parmar, P., Jay, P., Tanwar, S., Raboaca, M. S., et al. (2022). Stochastic neural networks-based algorithmic trading for the cryptocurrency market. *Mathematics*, 10(9):1456.

Kearns, M. and Nevmyvaka, Y. (2013). Machine learning for market microstructure and high frequency trading. In *High Frequency Trading: New Realities for Traders, Markets and Regulators*, pages 1–25. Risk Books.

Khaled, C., Hirchoua, B., Lmakri, A., et al. (2025). Optimizing deep reinforcement learning for financial markets: A comparative analysis of dqn, a2c, and ppo. In *International Conference on Digital Transformation*, pages 1–15. Springer.

Kim, S. and Kang, M. (2019). Financial series prediction using attention lstm. *arXiv preprint arXiv:1902.10877*.

Kurek, P. (2024). Algorithmic trading with long short-term memory recurrent neural network and stationary time series. Technical report, ResearchGate.

Kwon, D. H., Kim, J. B., Heo, J. S., Kim, C. M., et al. (2019). Time series classification of cryptocurrency price trend based on a recurrent lstm neural network. *Journal of Information Processing Systems*, 15(3):1–15.

Liu, L. and Si, Y. (2022). 1d convolutional neural networks for chart pattern classification in financial time series. *The Journal of Supercomputing*, 78:1–25.

Livieris, I. E., Pintelas, E., Stavroyiannis, S., and Pintelas, P. (2020). Ensemble deep learning models for forecasting cryptocurrency time-series. *Algorithms*, 13(5):121.

López de Prado, M. (2018). *Advances in financial machine learning*. John Wiley & Sons.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Mroziewicz, T. and Slepaczuk, R. (2026). A novel approach to trading strategy parameter optimization using double out-of-sample data and walk-forward techniques. *arXiv preprint arXiv:2602.10785*.

Ong, J. (2024). Modern portfolio construction with advanced deep learning models. Master's thesis, Singapore University of Technology and Design.

Prasetyo, R., Sumanto, S., Chaidir, I., et al. (2025). Reinforcement learning for bitcoin trading: A comparative study of ppo and dqn. *Jurnal Mandiri: Ilmu Pengetahuan, Seni, dan Teknologi*, 9(1):1–12.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Seabe, P. L., Moutsinga, C. R., and Pindza, E. (2023). Forecasting cryptocurrency prices using lstm, gru, and bi-directional lstm: a deep learning approach. *Fractal and Fractional*, 7(2):203.

Sezer, O. B. and Ozbayoglu, M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, 70:525–538.

Singh, J., Thulasiram, R., et al. (2022). Lstm based algorithmic trading model for bitcoin. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE.

Sirignano, J. A. (2019). Deep learning for limit order books. *Quantitative Finance*, 19(4):549–570.

Tanabe, H. and Senoguchi, J. (2025). Multi-timeframe fusion for bitcoin price direction prediction: An empirical study of temporal scale integration. Technical report, SSRN.

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., et al. (2020). Using deep learning for price prediction by exploiting stationary limit order book features. *Applied Soft Computing*, 93:106401.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

Vetrin, R. and Koberg, K. (2024). Reinforcement learning in optimisation of financial market trading strategy parameters. *Computer Research and Modeling*, 16(3):1–20.

Wang, J., Sun, T., Liu, B., Cao, Y., and Zhu, H. (2021). Clvsa: A convolutional lstm based variational sequence-to-sequence model with attention for predicting trends of financial markets. *arXiv preprint arXiv:2104.04041*.

Wen, X. and Li, W. (2023). Time series prediction based on lstm-attention-lstm model. *IEEE Access*, 11:1–15.

Wojarnik, G. (2023). The potential of convolutional neural networks for the analysis of stock charts. *Procedia Computer Science*, 219:1–10.

Yang, H., Liu, X., Zhong, S., and Walid, A. (2020). Deep reinforcement learning for automated stock trading: An ensemble strategy. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–8. ACM.

Zhang, Q., Qin, C., Zhang, Y., Bao, F., Zhang, C., et al. (2022). Transformer-based attention network for stock movement prediction. *Expert Systems with Applications*, 202:117239.

Zhang, X., Liang, X., Zhiyuli, A., Zhang, S., et al. (2019a). At-lstm: An attention-based lstm model for financial time series prediction. *IOP Conference Series: Materials Science and Engineering*, 569(5):052037.

Zhang, Z., Zohren, S., and Roberts, S. (2019b). Deeplob: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, 67(11):3001–3012.