Identification and Evaluation of Swing Points and True/False Breakout and Breakdown Signals

A Multi-Method Empirical and System-Based Study

Minze Wang

Last Updated: 2026

Contents

Chapter 1

Introduction and Problem Definition

Technical analysis relies heavily on the identification of price structures such as swing points, trendlines, and breakout or breakdown signals. However, a major challenge in practical trading systems is distinguishing **true signals** from **false signals**, which are often caused by short-term market noise.

This project focuses on three core tasks:

- **Task 1**: Study and compare methods for identifying swing points and differentiating true/false breakout and breakdown signals.
- **Task 2**: Implement at least two of these methods within an object-oriented (OOP) Python system.
- **Task 3**: Design and document a quantitative testing framework to evaluate the effectiveness of these methods using real and synthetic data.

The project adopts a balanced research–engineering approach, combining mathematical definitions, programmatic implementation, and empirical validation.

Chapter 2

Swing Point Identification Methods (Task 1)

2.1 Definition of Swing Points

Swing points form the foundation of trendline construction. A **Swing High** is a local maximum price, while a **Swing Low** is a local minimum price relative to neighboring observations. The quality of swing point identification directly impacts trendline accuracy and subsequent breakout detection.

2.2 Fixed Sliding Window Method

2.2.1 Principle

Given a closing price series

$$P = \{p_0, p_1, \ldots, p_{n-1}\},$$

a point p_t is classified as:

$$\text{Swing High if } p_t = \max\{p_{t-w}, \ldots, p_{t+w}\},$$

$$\text{Swing Low if } p_t = \min\{p_{t-w}, \ldots, p_{t+w}\},$$

where $w = 5$ and $t \in [w, n-1-w]$.

A minimum distance constraint is applied:

$$t_{i+1} - t_i \geq 3,$$

to filter noise-induced pseudo swing points.

2.2.2 Advantages and Limitations

- Advantages: simple, efficient ($O(n)$), easy to implement.
- Limitations: fixed window lacks adaptability across volatility regimes.

2.3 Volatility Adaptive Window Method

2.3.1 Principle

To improve adaptability, the window size is dynamically adjusted based on rolling volatility:

$$\sigma_t = \mathrm{std}(p_{t-19}, \dots, p_t), \quad \sigma_{\mathrm{avg}} = \frac{1}{n-19} \sum_{t=19}^{n-1} \sigma_t.$$

The adaptive window size is:

$$w_t = \max\left(3, \min\left(10, \mathrm{round}\left(5 \cdot \frac{\sigma_t}{\sigma_{\mathrm{avg}}}\right)\right)\right).$$

2.3.2 Advantages and Limitations

- Advantages: strong robustness across high- and low-volatility assets.
- Limitations: slightly higher computational complexity.

2.4 Slope Reversal Method

2.4.1 Principle

Let price differences be:

$$\Delta p_t = p_t - p_{t-1}.$$

Swing points are detected when the sign of Δp_t reverses:

$$\text{Swing High if } \Delta p_{t-1} > 0 \wedge \Delta p_t < 0,$$

$$\text{Swing Low if } \Delta p_{t-1} < 0 \wedge \Delta p_t > 0.$$

A minimum reversal amplitude threshold is applied to reduce noise.

Chapter 3

True/False Breakout and Breakdown Differentiation (Task 1)

3.1 Core Concept

A breakout or breakdown signal is considered a **candidate signal** when price crosses a trendline for two consecutive days. The key challenge is determining whether the signal represents a sustainable trend or a short-term fluctuation.

3.2 Holding Period and Threshold Method

3.2.1 Definition

Let p_s denote the price at signal time s. Over a holding period $H = 10$:

$$\text{True Breakout if } \frac{p_{\max}(s, s+H) - p_s}{p_s} \geq 3\%.$$

Breakdowns are defined symmetrically.

3.2.2 Strengths and Weaknesses

- Strengths: objective, reproducible, widely applicable.
- Weaknesses: fixed thresholds may underperform in extreme volatility.

3.3 Volume Confirmation Method

A valid breakout is more reliable when accompanied by abnormal trading volume:

$$V_s \geq 1.5 \times V_{20\text{avg}}.$$

This method is used as a secondary filter to improve precision.

3.4 Trendline Retest Validation

True signals often retest the original trendline (resistance becomes support, or vice versa) without breaking it again. Lack of a valid retest indicates pulse-type false signals.

Chapter 4

System Design and OOP Implementation (Task 2)

4.1 Overall Architecture

The system follows a modular object-oriented design:

- `DataFetcher`: data acquisition from Yahoo Finance, CSV, or synthetic generators.
- `SwingPointDetector`: swing point identification (fixed and adaptive).
- `TrendlineFitter`: OLS-based trendline estimation.
- `BreakoutConfirmer`: detection of candidate breakout/breakdown signals.
- `StockAnalyzer`: system orchestrator and analysis pipeline.

4.2 Implemented Methods

At least two approaches are implemented and compared:

- Fixed vs. adaptive swing point detection.
- Threshold-only vs. volume-confirmed breakout validation.

This design ensures high cohesion, low coupling, and extensibility.

Chapter 5

Testing Framework and Evaluation (Task 3)

5.1 Data Sources

- Real market data: AAPL, MSFT, GOOGL.
- Synthetic data: random walk and trend-plus-noise series.

5.2 Evaluation Metrics

- Swing point noise ratio.
- Precision, Recall, and F1-score for breakout signals.

5.3 Experimental Procedure

Each method is applied to all datasets, and results are compared to evaluate robustness and reliability.

Chapter 6

Results and Discussion

Empirical results show that:

- Adaptive window methods significantly reduce swing point noise.
- Volume confirmation improves breakout precision.
- No single method is optimal in all conditions.

Chapter 7

Conclusion and Future Work

This project demonstrates that combining adaptive swing point detection with multi-criteria breakout validation produces robust and reproducible results. Future work includes adaptive thresholds, volume-weighted models, and cross-asset validation.