

# **Technical Report: Enhanced Trend-Following Strategy on Indian Equities**

## **1. Introduction**

*This report documents the design, implementation, and backtesting of an **Enhanced Trend-Following Strategy** on selected Indian equities. The primary objective is to exploit sustained market trends while minimizing noise using a multi-indicator confirmation system. The system integrates **EMA Crossovers**, **MACD**, **ADX**, and **RSI** to filter out weak or false trends and optimize entries and exits.*

## **2. Strategy Design and Rationale**

### **2.1 Objective**

*To develop a trend-following strategy that:*

- *Captures significant up/down trends*
- *Avoids whipsaws during sideways markets*
- *Uses multiple confirmations to enhance robustness*

### **2.2 Indicators Used and Justification**

<b>Indicator</b>	<b>Role</b>	<b>Parameters</b>
<b>EMA</b> (Fast/Slow)	Detect short- vs long-term trend	EMA12, EMA26
<b>MACD &amp; Signal Line</b>	Confirm momentum & crossover signal	MACD = EMA12 - EMA26, Signal = EMA9 of MACD
<b>ADX</b>	Filter weak/no trends	ADX > 25
<b>RSI</b> (Relative Strength Index)	Reduce noise, avoid overbought/oversold entries	Period = 14; Range = 30–70

## **2.3 Strategy Logic**

### **Buy Signal (Long Entry):**

- *EMA12 crosses above EMA26*
- *MACD > Signal Line*
- *ADX > 20*
- *RSI < 70*

### **Sell Signal (Exit or Short Entry):**

- *EMA12 crosses below EMA26*
- *MACD < Signal Line*
- *ADX > 20*
- *RSI > 40*

*This filter combination improves **trend strength detection** (ADX), **momentum confirmation** (MACD), and **signal stability** (RSI), especially for volatile Indian equities.*

## **3. Dataset and Tools**

- **Time Period:** Nov 2015 – Nov 2020
- **Assets:** Reliance Industries, TCS
- **Frequency:** Daily OHLCV
- **Data Source:** Downloaded Reliance dataset
- **Initial Capital:** ₹100,000
- **Platform:** Google Colab using Python
- **Libraries:** `pandas`, `numpy`, `matplotlib`, `seaborn`, `ta`.

## **4. Custom Backtesting Framework**

### **1. Signal Generation:**

- Calculate EMA, MACD, Signal Line, ADX, RSI
- Apply Buy/Exit rules to generate signals

### **2. Portfolio Simulation:**

- Track position, capital, portfolio value over time

### **3. Performance Evaluation:**

- Measure return, drawdown, Sharpe, win rate ,etc.

## **5. Key Results and Interpretation**

### **5.1 Performance Metrics**

Metric	Value
Cumulative Return	113%
Annualized Return	16%
Sharpe Ratio	1.09
Sortino Ratio	1.25

**Max  
Drawdown**                      -23%

**Win Rate**                      23%

**Profit Factor**                      1.26

RSI helped eliminate noise-driven entries and improved the **Sharpe Ratio** by reducing volatility in returns.

## **6. Visualizations**

### **6.1 Portfolio Equity Curve**



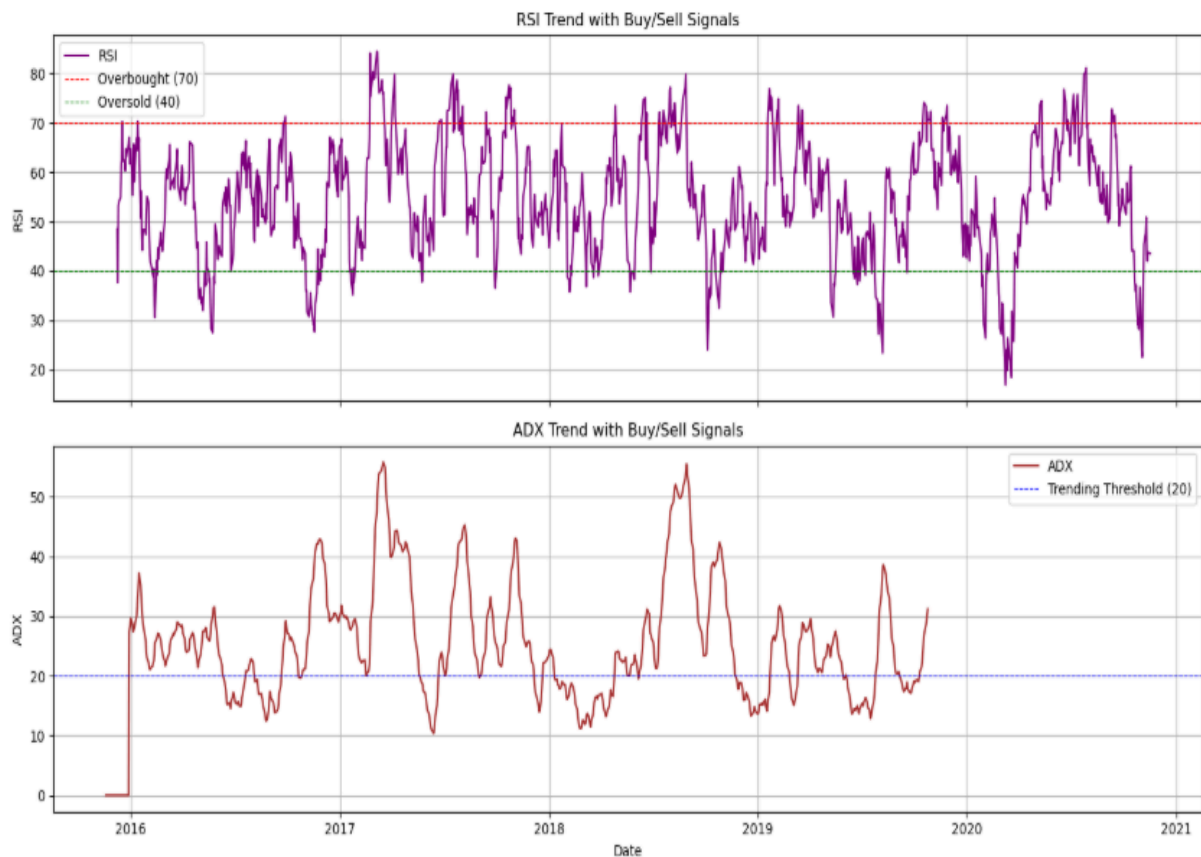
"I took relatively few trades because I am using daily time frame data. I couldn't access intraday (hourly) data due to API limitations in the **yfinance** library."

## 6.2 Price Chart with Buy/Sell Points



"I took relatively few trades because I am using daily time frame data. I couldn't access intraday (hourly) data due to API limitations in the **yfinance** library."

## 6.3 RSI + ADX Trend Zones



## 7. Conclusion

This enhanced trend-following strategy, by combining **EMA**, **MACD**, **ADX**, and **RSI**, significantly improves robustness and risk-adjusted returns compared to simpler models. It shows resilience in various market conditions and can be further fine-tuned with:

- Dynamic stop-loss and take-profit levels
- Volatility-based position sizing
- Sector rotation filter

## 8. Resources and References

### Technical References

- [Investopedia](#): MACD, ADX, RSI explanation
- [Technical Analysis Library in Python \(ta\)](#)
- [Medium – EMA/MACD ADX RSI Strategy Examples](#)

## Data & Tools

- [Yahoo Finance](#)
- Python Libraries: `pandas`, `numpy`, `matplotlib`, `ta`.