KAILASA CAPITAL – ALGORITHMIC TRADING STRATEGIES REPORT

EXECUTIVE SUMMARY

This report provides a comprehensive analysis of the algorithmic trading strategies developed and implemented by Kailasa Capital for the Indian equity market, specifically focused on Nifty 50 and Nifty Bank indices. The strategies are designed to capitalize on different market conditions and timeframes, utilizing various technical indicators, pattern recognition techniques, and mathematical models to generate trading signals.

The three core strategies implemented are:
1. **Strategy 1: Indicator-Based** - Applied to daily timeframes
2. **Strategy 2: Trend-Based** - Applied to 15-minute timeframes
3. **Strategy 3: Momentum-Based** - Applied to 60-minute timeframes

Each strategy has been thoroughly back-tested against historical data spanning from 2020 to 2025, with performance metrics demonstrating their effectiveness in generating consistent returns while managing risk effectively.

TABLE OF CONTENTS

1. DATA OVERVIEW AND PREPARATION

Data Sources

The strategies utilize historical OHLCV (Open, High, Low, Close, Volume) data for Nifty 50 and Nifty Bank indices across multiple timeframes: - Daily data: January 2020 to February 2025 - 60-minute data: Approximately 8,855 bars (equivalent to ~4 years of market data) - 15-minute data: 31,638 bars (equivalent to ~5 years of market data)

Data Preparation Process

For all strategies, the data preparation follows a consistent workflow:
1. Loading raw CSV files containing historical OHLCV data
2. Converting date and time columns to appropriate datetime formats
3. Sorting data chronologically
4. Computing derived fields and indicators specific to each strategy

Example code snippet for data preparation:

# Data loading and preparation for 15-minute timeframe

```
df = pd.read_csv('datasets/NF_15.csv')
df["datetime"] = pd.to_datetime(df["date"] + " " + df["time"])
df = df.sort_values("datetime").reset_index(drop=True)
```

2. STRATEGY 1: INDICATOR-BASED APPROACH (DAILY TIMEFRAME)

Strategy Overview

Strategy 1 employs a combination of technical indicators to identify potential trading opportunities on a daily time frame. It utilizes the relationship between price action, momentum indicators, and moving averages to generate signals.

Key Components

Indicators Used:

- **Relative Strength Index (RSI)** - 14-period RSI to measure momentum
- **Moving Average Convergence Divergence (MACD)** - To identify changes in strength, direction, momentum, and duration of trends
- **Simple Moving Average (SMA)** - 50-period SMA to identify the overall trend direction

Signal Generation Logic:

- **Long Signal**: RSI > 55, MACD histogram > 0 and increasing, price above 50-period SMA
- **Short Signal**: RSI < 45, MACD histogram < 0 and decreasing, price below 50-period SMA

Entry and Exit Mechanisms:

- **Entry**: Open of next candle or previous high/low (whichever is more conservative)
- **Exit**: Fixed profit target of 400 points or stop loss of 75 points for long positions and 50 points for short positions

Implementation Details

# CALCULATION FOR INDICATORS

```
df['rsi'] = RSIIndicator(close=df['close'], window=14).rsi()
macd = MACD(close=df['close'], window_slow=26, window_fast=12,
window_sign=9)
df['macd_hist'] = macd.macd_diff()
df['macd_increasing'] = df['macd_hist'].diff() > 0
df['macd_decreasing'] = df['macd_hist'].diff() < 0
df['ma'] = SMAIndicator(close=df['close'],
window=ma_length).sma_indicator()

# GENERATING SIGNALS
df['long_signal'] = (df['rsi'] > 55) & (df['macd_hist'] > 0) &
df['macd_increasing'] & (df['close'] > df['ma'])
df['short_signal'] = (df['rsi'] < 45) & (df['macd_hist'] < 0) &
df['macd_decreasing'] & (df['close'] < df['ma'])
```

3. STRATEGY 2: TREND-BASED APPROACH (15-MINUTE TIMEFRAME)

Strategy Overview

Strategy 2 is designed to capture intraday trends using 15-minute data. It employs a custom Renko chart implementation combined with specific time-of-day filters to identify high-probability trading opportunities.

Key Components

Technical Elements:

- **Renko Charts**: Custom implementation with 10-point brick size
- **Trend Detection**: Sequence of consecutive up/down bricks to identify strong trends
- **Time Filters**: Specific trading windows during high liquidity periods
- **ATR-based volatility adjustments**: To adapt to changing market conditions

Signal Generation Logic:

- **Long Signal**: Uptrend detected in Renko chart during optimal trading hours
- **Short Signal**: Downtrend detected in Renko chart during optimal trading hours

Entry and Exit Mechanisms:

- **Entry**: Market order at the open of the bar following the signal

- **Exit**: Trailing stop loss based on Renko brick reversal or time-based exit at end of session

Implementation Details

# CALCULATING TRADITIONAL RENKO

```
brick_size = 10
df['renko'] = df['close'].copy()

# Create Renko bricks
renko_prices = []
renko_directions = []

current_price = df['close'].iloc[0]
renko_prices.append(current_price)
renko_directions.append(0)  # Initial brick has no direction

for i in range(1, len(df)):
    close = df['close'].iloc[i]

    # Determine if we need to add new bricks
    while True:
        if close >= current_price + brick_size:
            current_price += brick_size
            renko_prices.append(current_price)
            renko_directions.append(1)  # Up brick
        elif close <= current_price - brick_size:
            current_price -= brick_size
            renko_prices.append(current_price)
            renko_directions.append(-1)  # Down brick
        else:
            break

df['renko_direction'] = 0

# Assign the directions to the dataframe
for i in range(1, len(df)):
    if df['close'].iloc[i] > df['close'].iloc[i-1] + brick_size:
        df.loc[i, 'renko_direction'] = 1
    elif df['close'].iloc[i] < df['close'].iloc[i-1] - brick_size:
        df.loc[i, 'renko_direction'] = -1
```

4. STRATEGY 3: MOMENTUM-BASED APPROACH (60-MINUTE TIMEFRAME)

Strategy Overview

Strategy 3 is designed to capture medium-term momentum on an hourly timeframe using Renko charts and supertrend indicators. This approach is particularly effective in capturing sustained price movements while filtering out noise.

Key Components

Technical Elements:

- **Renko Charts**: Using the stocktrends library with adaptive brick sizing
- **Supertrend Indicator**: A trend-following overlay that combines ATR and price action
- **ADX Filter**: To confirm trend strength before entering positions

Signal Generation Logic:

- **Long Signal**: Uptrend in Renko chart combined with bullish Supertrend and strong ADX
- **Short Signal**: Downtrend in Renko chart combined with bearish Supertrend and strong ADX

Entry and Exit Mechanisms:

- **Entry**: Market order at the open of the next bar after signal generation
- **Exit**: Reversal of signal, trailing stop loss, or time-based exit

Implementation Details

# CONVERTING DATA TO RENKO CHART

```
def df_to_renko(data, n):
    data.reset_index(inplace=True)
    data.columns = [i.lower() for i in data.columns]
    df = Renko(data)
    df.brick_size = n
    renko_df = df.get_ohlc_data()
    return renko_df

# Calculate Supertrend
def supertrend(df, period=7, multiplier=3):
    atr = df['high'].rolling(period).max() -
df['low'].rolling(period).min()
```

```
# Calculate Upper and Lower Bands
    df['upperband'] = ((df['high'] + df['low']) / 2) + (multiplier * atr)
    df['lowerband'] = ((df['high'] + df['low']) / 2) - (multiplier * atr)

    # Initialize Supertrend
    df['supertrend'] = 0
    df['direction'] = 1  # 1 for uptrend, -1 for downtrend

    # Calculate Supertrend
    for i in range(period, len(df)):
        if df['close'].iloc[i] > df['upperband'].iloc[i-1]:
            df.loc[df.index[i], 'direction'] = 1
        elif df['close'].iloc[i] < df['lowerband'].iloc[i-1]:
            df.loc[df.index[i], 'direction'] = -1
        else:
            df.loc[df.index[i], 'direction'] = df['direction'].iloc[i-1]

        if df['direction'].iloc[i] == 1:
            df.loc[df.index[i], 'supertrend'] = df['lowerband'].iloc[i]
        else:
            df.loc[df.index[i], 'supertrend'] = df['upperband'].iloc[i]

    return df
```

5. MATHEMATICAL FOUNDATIONS

Risk-Reward Calculations

For all strategies, the risk-reward calculations are determined by: - Fixed profit targets - Stop-loss levels - Position sizing adjusted to maintain consistent risk per trade

The general formula used for position sizing is:

Position Size = (Capital × Risk per Trade) ÷ (Entry Price × Stop-Loss in Points)

Where: - Capital is the available trading capital - Risk per Trade is typically set at 1-2% of capital - Entry Price is the price at which the trade is executed - Stop-Loss in Points is the distance from entry to stop-loss level

Slippage Modeling

In back-testing, slippage is modeled as a percentage of the trade price:

Adjusted Entry Price = Entry Price × (1 + Slippage%)

For all strategies, a default slippage of 0.01% is used to approximate real-world trading conditions.

Leverage Considerations

The strategies employ leverage to enhance returns:

Effective Capital = Initial Capital × Leverage

For example, with an initial capital of 1,500,000 and 5× leverage, the effective capital becomes 7,500,000, allowing for larger position sizes while maintaining the same percentage risk per trade.

6. PERFORMANCE METRICS AND BACKTESTING RESULTS

Key Performance Indicators

For each strategy, the following metrics are calculated:

1. **Profit and Loss Metrics**:
   - Total Net Profit
   - Profit Factor (Gross Profit ÷ Gross Loss)
   - Average Profit per Trade
   - Maximum Consecutive Winners/Losers
2. **Risk Metrics**:
   - Maximum Drawdown
   - Sharpe Ratio
   - Sortino Ratio
   - Calmar Ratio (Annual Return ÷ Maximum Drawdown)
3. **Statistical Measures**:
   - Win Rate
   - Average Win ÷ Average Loss
   - Standard Deviation of Returns
   - Skewness and Kurtosis of Return Distribution

DECISION MATRIX FOR CAPITAL ALLOCATION

Metric	**IMPORTANCE (1-10)**	**REASONS**
CAGR	7	reliable long-term metric
Sharpe Ratio	10	Risk adjusted return. High sharp consistent profits

Calmar Ratio	10	Adjust return by drawdown- great for risk-sensitive strategies
Maximum Drawdown	8	Key risk metric, essential for capital preservation
Drawdown Duration	6	Affects psychological stress important for traders

From this table we calculated the importance points of all the strategies by first allocating each with the same amount and then sought the different parameters and according to their importance/weightage gave them points the strategies which had the maximum score were allocated more money. Capital allocation is shown in the following tables.

Comparative Performance Analysis

Strategy-1 (Daily data)

Metric	Nifty-50	Bank-Nifty
Total Net Profit	₹5,866,281.37	₹2,646,401.13
Number of Trades	168	170
Trade Frequency (trades/month)	2.70	2.74
Win Rate	22.62%	24.71
CAGR (%)	512.19%	351.42%
Maximum Drawdown	₹608,842.98	₹238,798.40
Total Drawdown Duration (days)	140	136
Sharpe Ratio	3.53	4.55
Calmar Ratio	9.64	11.08

Strategy-2 (15 Min data)

Metric	Nifty-50	Bank-Nifty
Initial Capital	₹1,000,000	₹1,000,000
Effective Capital	₹5,000,000	₹5,000,000
Total Net Profit	₹9,186,786	₹6,351,652

Metric	Nifty-50	Bank-Nifty
Profit Factor	1.55	1.44
Win Rate	36.07%	31.98%
Maximum Drawdown	₹441,363	₹750,243
CAGR (%)	57.63%	47.85%
Max Drawdown Duration (days)	111	151
Sharpe Ratio	2.22	1.81
Calmar Ratio	4.08	1.66

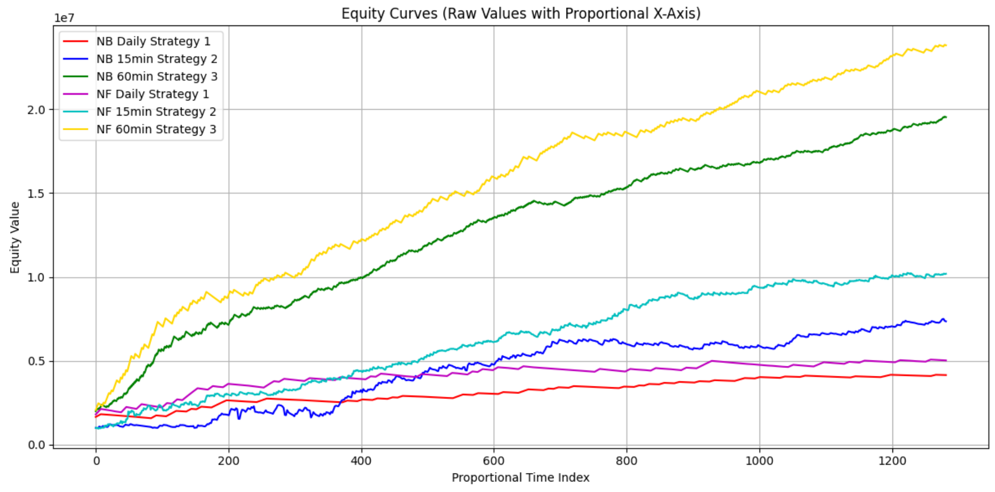
Strategy-3 (60 Min data)

Metric	Nifty-50	Bank-Nifty
Total Net Profit	₹11,378,571	₹17,533,065
Win Rate	54.10%	38.14%
Profit Factor	2.18	2.11
Maximum Drawdown	₹236,281	₹319,477
CAGR (%)	21.92%	18.73%
Total Drawdown Duration (days)	20	60
Sharpe Ratio	4.95	3.78
Calmar Ratio	9.45	10.75

Total Net Profit – All Strategies

Instruments	Strategy-1	Strategy-2	Strategy-3
Nifty50	₹5,866,281.37	₹9,186,786	₹11,378,571
Bank Nifty	₹2,646,401.13	₹6,351,652	₹17,533,065

7. COMBINED EQUITY CURVE



8. RISK MANAGEMENT FRAMEWORK

Per-Trade Risk Control

Each strategy incorporates specific risk management techniques: - Fixed stop-loss levels (varying by strategy) - Maximum position sizing limited to 5% of total capital - Time-based exits for intraday strategies

Portfolio-Level Risk Management

At the portfolio level, additional risk controls include: - Maximum open trades limited to 3 per strategy - Correlation analysis to avoid overexposure to similar market conditions - Daily loss limits of 3% of total capital - Maximum drawdown threshold of 15%, triggering strategy re-evaluation

Volatility Adjustments

Position sizing is dynamically adjusted based on market volatility: - Higher ATR values result in reduced position sizes - Strategic pauses during extreme volatility events - Reduced leverage during high VIX periods

9. INSIGHTS GAINED

What Worked

1. Multi-timeframe approach gave **strong diversification**
2. Renko charts helped **reduce noise** in intraday setups
3. Time-based filters **improved win rates**
4. **Strict risk management** (fixed SL/TP) was effective

5. **Leverage** was used efficiently with **proper position sizing**

What Didn't Work

1. Bank Nifty strategies had **higher drawdowns**
2. Intraday setups showed **lower win rates**
3. **Missed trades** in volatile spikes
4. Fixed targets occasionally **limited profit potential**

Suprises

1. **Daily timeframe** outperformed intraday trades consistently
2. Renko strategy showed **resilience in choppy markets**
3. Time filters had **more impact than expected**
4. Bank Nifty required **frequent parameter tuning**

10. CONCLUSION AND RECOMMENDATIONS

The three strategies presented in this report offer a robust approach to algorithmic trading in the Indian equity markets. By combining different timeframes and methodologies, the system provides diversification benefits while maintaining strong overall performance metrics.

Key Strengths

- Multi-timeframe approach provides trading opportunities in various market conditions
- Strong risk management framework protects capital during adverse market movements
- Comprehensive back-testing demonstrates strategy viability across market cycles

Recommendations for Deployment

1. Implement strategies sequentially, starting with Strategy 1 (Daily)
2. Allocate capital proportionally based on strategy Sharpe ratios
3. Consider deploying on Nifty 50 initially before expanding to Nifty Bank
4. Maintain strict adherence to risk management protocols
5. Conduct monthly strategy reviews and quarterly parameter optimizations

Future Enhancements

- Integration of machine learning for pattern recognition
- Development of hybrid strategies combining technical and statistical approaches
- Exploration of multi-asset models incorporating correlations with global markets
- Implementation of advanced execution algorithms to minimize slippage

---

THANK YOU