3 Institutional-Grade Nifty Options Strategies for Your HFT Bot

Strategy 1: **Liquidity Sweep Reversal (Stop Hunt)**

The Logic: Why It Works

Market makers and algorithms intentionally push price **just beyond** key psychological levels (Previous Day High/Low, Round numbers, Swing Points) to trigger retail stop-losses and breakout orders. This creates a liquidity pool they can absorb. Once stops are triggered, there's no real demand, and price **snaps back violently**. You're fading the trap.

The Edge: Retail traders place stops 1-5 ticks beyond obvious levels. Smart money knows this. The "wick rejection" is your signal that liquidity has been absorbed.

The Setup (The "If")
```

1. Identify Key Level: Previous Day High (PDH) or Previous Day Low (PDL)
OR Round number (e.g., Nifty 21,500 / 21,000)

2. Price Action Requirement:
- Price BREAKS the level by 5-20 points (the "sweep")
- BUT the **5-minute candle closes BACK INSIDE** the level (rejection)

3. Volume Spike Confirmation:
- The breakout candle has volume > 1.5x the 20-period average
(This confirms stop-hunt activity)

4. Time Filter:
- Between 9:30 AM - 3:00 PM (avoid first 15 mins and last 30 mins)
```

The Trigger (The "Then")

For CALL Options (Bullish Reversal from PDL Sweep):
```

ENTRY:
- Buy ATM/Slightly OTM CALL when:
```

* Next candle opens ABOVE the key level (confirms rejection)

* OR price reclaims VWAP if it was below

STOP LOSS:

- Below the "sweep low" (the lowest wick point) - typically 15-25 points

TARGET:

- 1:2 Risk-Reward initially

- Trail SL to breakeven when price moves 1R in your favor
```

**For PUT Options (Bearish Reversal from PDH Sweep):**
```

ENTRY:

- Buy ATM/Slightly OTM PUT when:

* Next candle opens BELOW the key level

STOP LOSS:

- Above the "sweep high" + 15-25 points

TARGET:

- 1:2 R:R, trail to breakeven at 1R
```

---

**Pseudocode Logic**

```python

# Strategy 1: Liquidity Sweep Reversal

# --- INPUTS ---

PDH = previous_day_high()
PDL = previous_day_low()
current_time = get_current_time()
current_price = get_nifty_spot()
candle_5m = get_candle_data(timeframe='5m')

# --- CALCULATE ---

vol_avg_20 = average(volume, period=20)
candle_high = candle_5m['high']
```

```python
candle_low = candle_5m['low']
candle_close = candle_5m['close']
candle_volume = candle_5m['volume']
```

--- BEARISH SETUP (PDH Sweep) ---

```python
if current_time >= '09:30' and current_time <= '15:00':

    # Check if price swept ABOVE PDH
    if candle_high > PDH + 5 and candle_high < PDH + 20:

        # Check if candle CLOSED back below PDH (rejection)
        if candle_close < PDH:

            # Volume confirmation
            if candle_volume > 1.5 * vol_avg_20:

                # TRIGGER: Enter PUT on next candle open
                if next_candle_open() < PDH:
                    BUY_PUT(strike='ATM')
                    stop_loss = candle_high + 20  # Above sweep high
                    target = entry_price - (2 * (stop_loss - entry_price))

                    print("LIQUIDITY SWEEP DETECTED - Entering PUT")
```

--- BULLISH SETUP (PDL Sweep) ---

```python
if current_time >= '09:30' and current_time <= '15:00':

    if candle_low < PDL - 5 and candle_low > PDL - 20:

        if candle_close > PDL:

            if candle_volume > 1.5 * vol_avg_20:

                if next_candle_open() > PDL:
                    BUY_CALL(strike='ATM')
                    stop_loss = candle_low - 20
                    target = entry_price + (2 * (entry_price - stop_loss))

                    print("LIQUIDITY SWEEP DETECTED - Entering CALL")
```

Strategy 2: **Failed Opening Range Breakout (ORB Fakeout)**

The Logic: Why It Works
The first 15-30 minutes establish the "Opening Range" (OR). Breakout traders aggressively buy/sell the first break of this range. **But 60-70% of first breakouts FAIL** because:
- Low liquidity in the first 30 mins creates volatile wicks
- Algos test both sides before committing to direction
- Retail FOMO enters at the worst prices

You're fading the amateur breakout traders when they get trapped.

The Setup (The "If")

```

1. Define Opening Range (OR):
- High and Low of first 15 minutes (9:15 AM - 9:30 AM)
- Store: OR_High, OR_Low

2. Wait for Initial Breakout:
- Price breaks ABOVE OR_High OR BELOW OR_Low
- Time: Between 9:30 AM - 10:30 AM

3. Fakeout Confirmation (Critical):
- Breakout candle has a LONG WICK (wick > 60% of candle body)
- Price closes BACK inside OR within 2 candles
- Volume on breakout candle is LOWER than previous 5-candle average
(Low volume = weak breakout = fakeout likely)

4. Additional Filter:
- Check if Nifty is trading NEAR VWAP (±30 points)
- This indicates consolidation, making fakeout more reliable
```

The Trigger (The "Then")

For CALL Entry (Fakeout Below OR_Low):
```

ENTRY:

- Buy ATM CALL when:

* Price re-enters OR and closes above OR_Low

* Next candle opens inside OR

STOP LOSS:

- 20 points below the fakeout wick low

TARGET:

- OR_High (the opposite side of the range)

- OR OR_High + 50 points if momentum is strong
```


**For PUT Entry (Fakeout Above OR_High):**
```

ENTRY:

- Buy ATM PUT when:

* Price re-enters OR and closes below OR_High

STOP LOSS:

- 20 points above the fakeout wick high

TARGET:

- OR_Low
```


---


**Pseudocode Logic**

```python

# Strategy 2: Failed Opening Range Breakout

# --- DEFINE OR ---

OR_start = '09:15'
OR_end = '09:30'
OR_high = max(price[OR_start:OR_end])
OR_low = min(price[OR_start:OR_end])
```

--- MONITOR FOR BREAKOUT ---

```
current_time = get_current_time()
candle = get_candle_data(timeframe='5m')
vwap = calculate_vwap()

if current_time >= '09:30' and current_time <= '10:30':

# --- BEARISH FAKEOUT (Price broke above OR_High) ---
if candle['high'] > OR_high:

# Check for long wick (wick > 60% of body)
body_size = abs(candle['close'] - candle['open'])
upper_wick = candle['high'] - max(candle['close'], candle['open'])

if upper_wick > 0.6 * body_size:

# Check if closed back inside OR
if candle['close'] < OR_high:

# Volume check (weak breakout)
vol_avg5 = average(volume, period=5)
if candle['volume'] < vol_avg5:

# VWAP proximity check
if abs(current_price - vwap) < 30:

# TRIGGER: Enter PUT
if next_candle_open() < OR_high:
BUY_PUT(strike='ATM')
stop_loss = candle['high'] + 20
target = OR_low

 print ("FAKEOUT - Entering PUT")

# --- BULLISH FAKEOUT (Price broke below OR_Low) ---
if candle['low'] < OR_low:

body_size = abs(candle['close'] - candle['open'])
lower_wick = min(candle['close'], candle['open']) - candle['low']

if lower_wick > 0.6 * body_size:
```

```
if candle['close'] > OR_low:

volavg5 = average(volume, period=5)
if candle['volume'] < volavg5:

if abs(current_price - vwap) < 30:

if nextcandleopen() > OR_low:
BUY_CALL(strike='ATM')
stop_loss = candle['low'] - 20
target = OR_high


 p r i n t (O8-BJ5AKEOUT - Entering CALL")
```

Strategy 3: **VWAP Extreme Deviation with Tape Reading (Mean Reversion)**

The Logic: Why It Works
VWAP (Volume Weighted Average Price) is the **institutional benchmark**. When price stretches >2 standard deviations from VWAP, it's **statistically overstretched**. Institutions will fade this move because:
- They need to execute at favorable prices (near VWAP)
- Algos programmed to buy dips to VWAP / sell rips away from VWAP
- Retail exhaustion happens at extremes

The Tape Reading Edge: You confirm the reversal by watching for **large limit orders absorbing market orders** at the extreme.

The Setup (The "If")

```

1. Calculate VWAP and Standard Deviation Bands:
- VWAP = Cumulative(Price × Volume) / Cumulative(Volume)
- StdDev = Standard deviation of (Price - VWAP) over last 50 candles
- Upper Band = VWAP + (2 × StdDev)
- Lower Band = VWAP - (2 × StdDev)

2. Price Extreme:

- Price touches or crosses Upper Band (for PUT setup)

- Price touches or crosses Lower Band (for CALL setup)


3. Time Filter:

- Between 10:00 AM - 2:30 PM (avoid opening/closing volatility)


4. Tape Reading Confirmation (THIS IS KEY):

- Check Order Book Depth:

* For PUT: Large BID SIZE appears near the extreme (institutions absorbing sellers)

* For CALL: Large ASK SIZE appears near the extreme (institutions absorbing buyers)


- OR: Price "stalls" at the extreme for 2+ candles (cannot push further)
```

The Trigger (The "Then")

For PUT Entry (Price above Upper Band):
```

ENTRY:

- Buy ATM PUT when:

* Price is > VWAP + 2×StdDev

* Next 5-min candle closes BELOW the Upper Band (rejection)

* OR large bid size detected in order book

STOP LOSS:

- Above the extreme high + 1.5×StdDev (approximately 40-50 points)

TARGET:

- VWAP itself (mean reversion target)

- Partial exit at VWAP + 1×StdDev
```


### For CALL Entry (Price below Lower Band):
```

ENTRY:

- Buy ATM CALL when:

* Price is < VWAP - 2×StdDev

* Next candle closes ABOVE the Lower Band

STOP LOSS:
- Below the extreme low - 1.5×StdDev

TARGET:
- VWAP
```

---

**Pseudocode Logic**

```python
```

Strategy 3: VWAP Extreme Deviation Mean Reversion

import numpy as np

--- CALCULATE VWAP AND BANDS ---

def calculate*vwap*bands():
cumulative_pv = cumsum(price * volume)
cumulative_volume = cumsum(volume)
vwap = cumulative*pv / cumulative*volume

# Calculate standard deviation
price_deviation = price - vwap
std*dev = np.std(price*deviation[-50:]) # Last 50 candles

upper*band = vwap + (2 * std*dev)
lower*band = vwap - (2 * std*dev)

return vwap, upper*band, lower*band, std_dev

--- MAIN LOGIC ---

vwap, upper*band, lower*band, std*dev = calculate*vwap_bands()
current*price = get*nifty_spot()
current*time = get*current_time()
candle = get*candle*data(timeframe='5m')

Optional: Get order book data (requires

Level 2 data access)

```python
order_book = get_order_book()  # Returns bid/ask sizes

if current_time >= '10:00' and current_time <= '14:30':

    # --- BEARISH SETUP (Price above Upper Band) ---
    if current_price > upper_band:

        # Check if candle closed back below upper band (rejection)
        if candle['close'] < upper_band:

            # Tape reading: Check for large bids (optional if you have data)
            # If order_book['total_bid_size'] > 2 * average_bid_size:

            # OR: Check if price stalled (2 consecutive candles at extreme)
            prev_candle = get_candle_data(timeframe='5m', offset=1)
            if prev_candle['high'] >= upper_band:

                # TRIGGER: Enter PUT
                BUY_PUT(strike='ATM')
                stop_loss = candle['high'] + (1.5 * std_dev)
                target = vwap

                print(f"🔴 VWAP EXTREME DEVIATION - Entering PUT")
                print(f"   Entry: {current_price}, Target: {vwap}, SL: {stop_loss}")

    # --- BULLISH SETUP (Price below Lower Band) ---
    if current_price < lower_band:

        if candle['close'] > lower_band:

            # Check for stall or large asks
            prev_candle = get_candle_data(timeframe='5m', offset=1)
            if prev_candle['low'] <= lower_band:

                BUY_CALL(strike='ATM')
                stop_loss = candle['low'] - (1.5 * std_dev)
                target = vwap

                print(f"🟢 VWAP EXTREME DEVIATION - Entering CALL")
                print(f"   Entry: {current_price}, Target: {vwap}, SL: {stop_loss}")
```

--- TRAIL STOP LOGIC ---

```
if in_position:
if position_type == 'PUT':
if current_price <= vwap + std_dev:  # Halfway to target
stop_loss = entry_price  # Move to breakeven

if position_type == 'CALL':
if current_price >= vwap - std_dev:
stop_loss = entry_price
```

Ø=Ý%Implementation Tips for Your Bot**

Data Requirements:
1. **5-minute candle data** with Open, High, Low, Close, Volume
2. **Previous Day High/Low** (store at EOD)
3. **Real-time VWAP calculation** (cumulative)
4. **(Optional but powerful):** Level 2 Order Book data for tape reading

Risk Management (CRITICAL):
```python
```
Never risk more than 1-2% per trade

```
account_size = 100000  # Your capital
risk_per_trade = 0.02  # 2%
max_loss_per_trade = account_size * risk_per_trade  # [1] 2 0 0 0
```

Calculate position size based on stop loss

```
stop_loss_points = 30  # Example
position_size = max_loss_per_trade / stop_loss_points
```

Backtesting Before Live:
- Test on **at least 6 months** of historical 5-min Nifty data
- Calculate:
- Win Rate (should be >50% for these strategies)
- Average R:R (should be >1.5:1)
- Max Drawdown (should be <15%)

Time-Based Filters (Important for Nifty):
- **Avoid 9:15-9:30:** Opening volatility, fake moves

- **Best Time:** 10:00 AM - 2:30 PM (institutional hours)
- **Avoid 3:15-3:30:** Closing volatility, unpredictable

& þ**Critical Risk Warnings**

These strategies are **NOT guaranteed** to work. Options trading, especially HFT, carries **significant risk of total capital loss**. Before deploying:

1. **Paper trade for 1-2 months** minimum
2. **Start with SMALL position sizes** (¹ 5 0 0 - 1 per trade)
3. **Never trade without stop losses**
4. **Understand that past patterns may not repeat**
5. Options decay (Theta) works against you—these are **intraday strategies only**

These strategies exploit **real market microstructure inefficiencies**, but markets adapt. What works today may stop working in 6 months. Continuously monitor performance and be ready to adapt. Good luck building your bot!