Volatility-Aware Multi-Market Strategy

Vishwa Kalal

Core Concepts

This strategy applies a single, systematic logic consistently across a set of independent markets, including equities, cryptocurrencies, and the foreign exchange market. The choice of a multi-asset approach was an intentional decision made to leverage the benefits of diversification, since factors that drive volatility in each asset are often unique, for example, a new crypto regulation is unlikely to have a major impact on equity markets or the foreign exchange market. By applying the same logic across the different markets, the strategy spreads out its risk, which reduces the chance that one catastrophic event will cause the model to fail.

The strategy's architecture consists of three primary components:

1. **Volatility Measurement:** The Average True Range (ATR) indicator serves as an objective measure of market volatility.
2. **Signal Generation:** The Supertrend indicator identifies the primary trend direction, which is then confirmed by two filters: the price must trade above its 100-day Exponential Moving Average (EMA), and the Relative Strength Index (RSI) must be above 55.
3. **Risk Management:** A volatility-weighted position sizing, which dynamically adjusts trade exposure to maintain a constant level of risk per trade.

Economic Rationale

The economic hypothesis of this strategy is that market volatility presents both a risk and an opportunity. My approach is designed to exploit this duality for profit. The market is not perfectly efficient; predictable patterns driven by investor behavior emerge during different volatility periods. This strategy is built to capitalize on these patterns, specifically the trend momentum that follows these volatility spikes. Importantly, the alpha is not generated by predicting price direction, but through risk management. The strategy captures confirmed trends by proportionally adjusting its risk exposure to an asset's recent volatility. This process normalizes risk across the whole portfolio by allocating smaller positions to more volatile assets and larger positions to less volatile assets. This turns volatility from an unmanaged threat into a key input for determining positions. The result is focusing on the trend following risk more efficiently, which aims for smoother, risk-adjusted returns in the long term.

Detailed Implementation

Architectural Framework: The strategy is implemented within a systematic, rules-based architectural framework. The model operates across three different markets: Cryptocurrencies, Equities, and Forex, to capitalize on diversification benefits. The selection of these assets provides exposure to different macroeconomic drivers and market behaviors. The algorithm's design works across any time frame, but it was tested on daily bars to focus on significant, medium-term trends, while filtering out short-term price fluctuations.

Volatility Measurement: The strategy's foundation for determining volatility is the Average True Range (ATR) indicator. ATR is a technical analysis tool that measures how much an asset's price moves. It calculates volatility by looking at the "true range" of an asset's price during a specific period of time, which accounts for gaps and other price fluctuations. Essentially, it gives the average amount of price movement during a timeframe.

The calculation begins with the True Range (TR), which is defined as the greatest of the following:

- Current Period High - Current Period Low
- | Current Period High - Previous Period Close |
- | Current Period Low - Previous Period Close |

The ATR is then calculated as a 14-day period moving average of the TR values. The result is an objective, quantification of an asset's recent price price fluctuation. An expanding ATR indicates increasing volatility, while a contracting ATR indicates a decreasing volatility. This metric serves as the main input for trend-identification signal and risk management overlay.

Trend Identification: The main source of generating trade signals is the Supertrend indicator, a trend following tool made by Olivier Seban. I chose this indicator because it engages in the principle of volatility awareness at the most important part of the algorithm. The Supertrend indicator uses the ATR to plot dynamic upper and lower bands around the price of an asset, creating a single line that follows the action of the line.

This calculation for the bands are:
- Upper Band = (High + Low)/2 + Multiplier * ATR
- Lower Band = (High + Low)/2 - Multiplier * ATR

A trade signal is created when the closing price crosses one of these bands. When the price closes above the band, the indicator line flips below the price and turns green, signaling an uptrend. Similarly, when the price closes under the band, the indicator line flips above the price and turns red, signaling a downtrend.

Unlike simple moving average crossovers, this indicator uses ATR to dynamically adapt its bands to the market. During periods of high volatility, the band widens to reduce false signals in erratic markets. On the contrary, during periods of low volatility, the bands tighten to make the indicator more sensitive to the start of new trends.

Trade Execution Logic: The strategy's execution framework is set up by a multi-stage set of rules. It is a long-only strategy and does not engage in short selling.
- **Entry Rule:** A long position is initiated when four conditions prove a strong bullish trend (*Figure 5*):
    1. The price is above two Supertrend indicators (2.5, 3)
    2. The RSI is greater than 55, indicating strong buying power
    3. The price is greater than the 100-day EMA, confirming a long-term uptrend
    4. The price must be below a factor (1.10 for crypto, 1.02 for stocks) of the 5-day Time-Weighted Average Price (TWAP), to avoid chasing overbought moves
- **Exit Rule:** The strategy's exit condition is triggered when the price closes below both Supertrend indicator lines. This functions as a dynamic stop-loss feature. As the price trends upward, the Supertrend lines trail below it, progressively locking in profits and making sure every trade has a volatility-adjusted exit point.

Risk Assessment

Market Volatility Risk: Different markets carry different volatility risks, for instance, a position in a highly volatile asset like Bitcoin (BTC) carries much greater dollar-risk than a position of a less volatile stock like Apple (AAPL). On any day, BTC can move 5-10%, whereas AAPL might only move 1-2%. A static approach ignores this, leading to an unstable risk profile.
- **Mitigant:** The strategy sizes each position to be inversely proportional to its ATR. In practice, if the algorithm determined that BTC was five times more volatile than AAPL, it would allocate only 5% of the portfolio to a position in BTC while allocating 20% of the portfolio into AAPL. (*Figure 4*)

Systemic Market Risk: During a broad market downturn, most assets, especially equities, tend to become highly correlated and decline together. Most strategies ignore the macroeconomic state which exposes them to major drawdowns.

- **Mitigant:** This strategy incorporates a regime filter by monitoring the S&P 500 (SPY) and compares its price to its 200-day EMA. If SPY is trading below this moving average, the algorithm enters a "risk-off mode" and can no longer start any long trades in equities.

Model Overfitting Risk: The strategy's parameters may be over-optimized to the specific historical data used in the backtest. This could create a model that looks perfect on past data but fails to adapt to live market conditions.

- **Mitigant:** Rather than relying on complex rules, the model is built on well established, and economically-backed principles like trend-following (Supertrend), and momentum (RSI). These concepts have been historically sound across different market cycles.

Capital and Liquidity

This strategy is engineered to manage large amounts of capital, which is supported by its strong performance and low trading costs. The backtests have produced an estimated strategy capacity of $60,000,000, a direct result of its ability to use capital and manage liquidity well.

High Capital Capacity: The strategy's ability to scale is driven by its low trading frequency. Over a 6-year backtest period (2019-2025), it executed only 214 trades, resulting in a very low Portfolio Turnover of 1.15%. This represents the strategy's patient, long-term approach, which is important for scalability as it naturally minimizes market impact and prevents performance decline.

Efficient Liquidity Management: The strategy's cost-effectiveness is direct proof of its efficiency in liquid markets. Despite trading a total volume of over $6.6M, it only paid $334.23 in fees, showing minimal slippage and tight bid-ask spreads.

Liquidity Constraint: The backtests identified Ethereum (ETH) as the "Lowest Capacity Asset", providing a clear, data-backed understanding of the main liquidity drawback, which allows for risk management as the strategy scales.

Analytical Evidence

The strategy was backtested from January 1, 2019 to January 1, 2025. This period was chosen to test the model across different market regimes, including the COVID-19 crash, followed by a bullish market, and periods of high inflation. To ensure reliability, the strategy applied the same rules to all assets, which prevents common errors like overfitting the strategy to past performance.

Key Performance Metrics: The results showed a very favorable risk-return profile. The strategy generated a Compound Annual Growth Rate (CAGR) of 22.5% with a Sharpe Ratio of 1.27. This combination is important: the high return rate represents the strategy's raw profitability, while the Sharpe Ratio proves that the returns were made efficiently. Together they back the strategy's main design, which is built to manage volatility while maximizing returns per risk taken. (*Figure 1*)

Alpha and Beta Analysis: Alpha and Beta are the most important metrics because they prove the strategy achieved its goal: high-quality returns that are independent of the broader market. The backtests produced an extremely low Beta of 0.158, this metric shows us that the portfolio's performance is largely uncorrelated with the S&P 500. This low Beta was not accidental, it is a direct effect of the market regime filter that disconnects the strategy from market downturns. The strong Alpha value of 0.112 (11.2%) represents the strategy's excess return generated by its unique logic, independent of favorable market conditions. This level of Alpha demonstrates the model's ability to produce valuable returns. (*Figure 2*)

Performance Metrics		Risk Metrics		Trade Statistics		Cost & Capacity	
CAGR	22.50%	Alpha	0.112	Win Rate	51%	Total Fees	$334.23
Sharpe Ratio	1.267	Beta	0.158	Profit-Loss Ratio	4.77	Portfolio Turnover	1.15%
Sortino Ratio	1.371	Annual Standard Deviation	0.101	Average Win	2.27%	Estimated Capacity	$60M
Net Profit	268.32%	Maximum Drawdown	10.70%	Average Loss	-0.48%	Lowest Capacity Asset	ETH

Figure 1. Backtest results summary

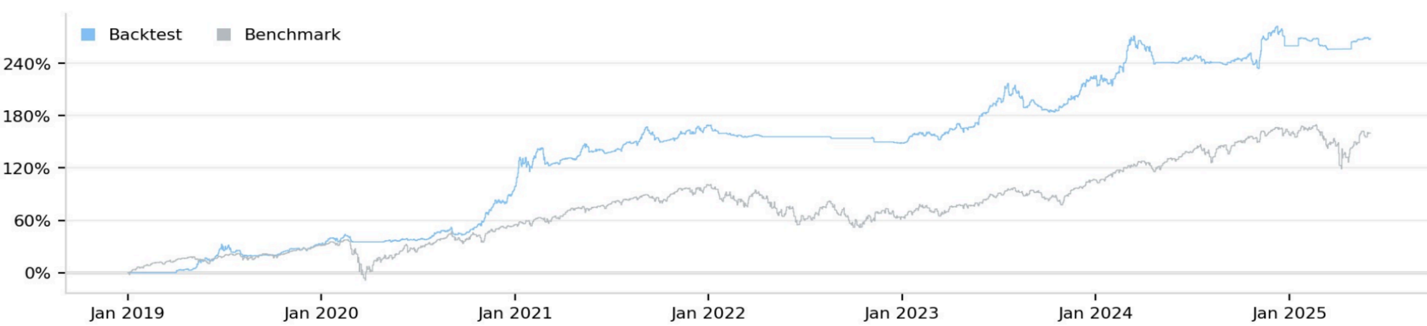

Figure 2. Strategy vs. S&P 500

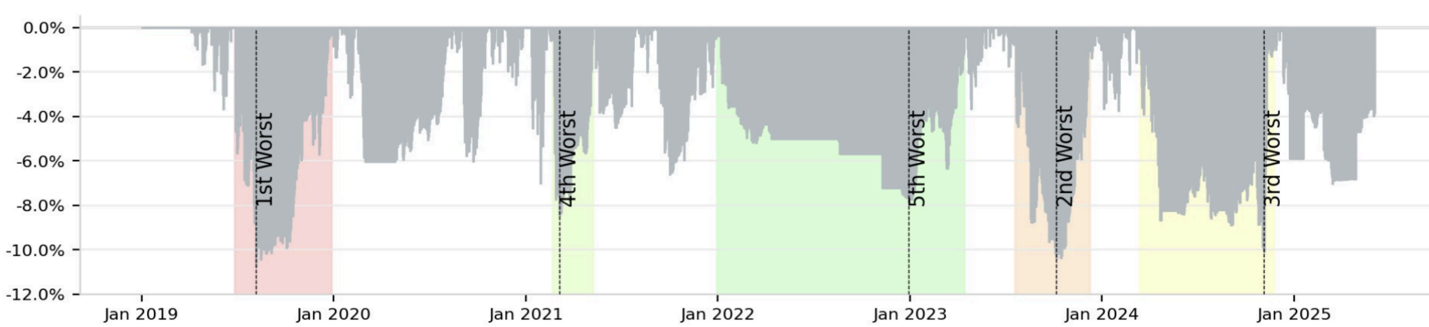

Figure 3. Strategy Drawdown (Jan 2019 - Jan 2025)

```
def VolatilityWeight(self, symbol, price):
    atr_indicator = self.atr.get(symbol, None)
    if atr_indicator is None or not atr_indicator.IsReady or price <= 0:
        return None

    atr_value = atr_indicator.Current.Value
    if atr_value <= 0:
        return None

    daily_volume = atr_value / price
    annual_volume = daily_volume * sqrt(252)
    raw_weight = self.target_portfolio_volume / (annual_volume * 3.0)
    return max(self.min_weight, min(self.max_weight, raw_weight))
```

Figure 4. Volatility-Based Portfolio Allocation

```
# Entry condition
if symbol in self.stock_symbols:
    if (not self.spy_ema200.IsReady) or (self.Securities[self.spy_symbol].Price
                                          < self.spy_ema200.Current.Value):
        continue  # risk-off
if self.entry_prices[symbol] is None:
    if self.ExposureCap() >= self.exposure_cap:
        continue
    if (current_price > supertrend1 and
        current_price > supertrend2 and
        rsi > 55 and
        current_price > ema100 and
        current_price < factor * weekly_twap):  # Use appropriate factor
        self.Debug(f"{symbol}: Supertrend1={supertrend1}, Supertrend2={supertrend2}, RSI={rsi},
                    EMA100={ema100}, Weekly TWAP={weekly_twap}")
        #self.SetHoldings(symbol, 0.2)
        weight = self.VolatilityWeight(symbol, current_price)
        if weight is None:
            weight = 0.1
        self.SetHoldings(symbol, float(weight))
        self.entry_prices[symbol] = current_price
```

Figure 5. Entry Conditions