K-means Clustering Indicator

User Guide

ClickAlgo–FinWalt

Version 1.1.0 | May 30, 2025

May 30, 2025

Contents

1. Introduction

Cluster K-means Simplified is a high-performance, behaviour-based clustering indicator for **cTrader/cAlgo**. Leveraging ML.NET, it segments each bar into distinct market regimes—from directional trends to sideways consolidations—by combining Principal Component Analysis (PCA) for dimensionality reduction and K-Means for cluster assignment.

1.1. Why Clustering?

- **Capture market regimes**: Integrates 22 input features into cohesive regimes, revealing dominant behaviour.
- **Adapt to changing conditions**: Rolling-window retraining keeps the model current as market dynamics evolve.
- **Filter noise**: Groups similar patterns to avoid whipsaws and highlight genuine shifts.

1.2. How It Works

1. **Feature engineering (22 features)**
   - Price-based: log return, candle body, wicks.
   - Momentum: RSI, MACD histogram, ROC, CCI.
   - Trend strength: ADX, ADXR, DI^+.
   - Volatility: ATR, Bollinger Bands.
   - Volume: OBV, MFI, SMA volume, VWAP deviation.
2. **Normalise & PCA**: min–max normalisation followed by projection onto `PcaComponents` axes.
3. **K-Means clustering**: groups projected feature vectors into K clusters (user-defined $K = 2$–6).
4. **Cluster labelling & colouring**: ranks clusters by average pip delta; assigns bullish, bearish or range colours.
5. **Real-time metrics**:
   - **Davies–Bouldin Index (DB)**: measures average similarity between each cluster and its most similar one. Lower values (closer to 0) indicate well-separated, compact clusters. A DB < 0.5 is considered excellent; values < 1.0 are good.
   - **Silhouette Score (S)**: for each sample, computes cohesion vs. separation, ranging from -1 to $+1$. Scores above 0.5 indicate distinct clusters; 0.25–0.5 is fair; below 0 may imply overlapping clusters.
   - **Average Inter-Cluster Distance (ICD)**: the mean Euclidean distance between cluster centroids. Higher ICD reflects more distinct regimes. Aim for an ICD at least twice the average intra-cluster diameter.
6. **Rolling retraining**: after each `Retraining Interval` bars, retrain on the latest `Training History Bars` to maintain accuracy.

Interpreting Metrics A strong clustering configuration exhibits:

- **Low DB Index**: typically below 1.0, with values near 0.5 or less denoting highly separated clusters.
- **High Silhouette**: above 0.5 indicates clear separation; values above 0.7 are outstanding.
- **Substantial ICD**: when ICD significantly exceeds intra-cluster spread, regimes are well differentiated.

Key Benefits

- Clear visual filter of regimes.
- Systematic, data-driven classification.
- Optional Telegram alerts for regime changes.

2. Market Regime Detection Utility

Accurately distinguishing whether the market is *trending* or *ranging* is the cornerstone of indicator selection and risk management. The K-Means engine classifies every bar into a regime cluster in real time; this single signal can then be routed to any strategy component—entry logic, filters, position sizing, or alerts.

2.1. How Regime Signals Guide Indicator Choice

1. **Trending clusters**

   When the current bar is labelled as part of a *trend cluster* (high directional pip-delta, strong ADX projection), favour momentum-based tools:
   - Moving-average crossover, MACD line slope
   - Trend break-out filters (Donchian, SuperTrend, ATR-based stops)
   - Trailing-stop mechanisms that ride persistent directionality
2. **Ranging clusters**

   When the active cluster shows low net movement and higher inside-bar frequency, switch to mean-reversion logic:
   - Oscillators at extremes (RSI, Stochastic, CCI)
   - Volatility bands (Bollinger Bands, Keltner Channels) for fade entries
   - Fixed-target scalping with tight stops to harvest small oscillations

2.2. Key Benefits

- **Context-aware entries**: Prevents applying trend indicators in sideways markets and vice-versa, reducing whipsaws and false signals.
- **Adaptive risk profile**: Position size, stop distance, and profit target can auto-scale to the volatility and persistence typical of each cluster.
- **Objective rule switching**: Removes discretionary bias; regime transitions are data-driven and reproducible.
- **Strategy diversification**: Enables hybrid systems—trend-following during expansion phases and range-trading during consolidation—without manual intervention.
- **Performance stability**: Historical walk-forward tests show lower drawdowns and smoother equity curves when regime filtering is active.
- **Reduced over-fitting**: By segmenting the market into homogeneous blocks, parameter optimisation is performed within regimes, increasing robustness across unseen data.
- **Real-time alerts**: Telegram or local pop-ups can warn of imminent regime flips, giving discretionary traders time to shift tactics.

2.3. Implementation Tips

- Map each cluster ID to a `RegimeType` enum (e.g. `BullTrend`, `BearTrend`, `Range`) once, then reference that enum in all strategy modules.
- Test regime-specific parameter sets in isolation before combining them into a unified strategy.
- For lower time-frames (<M15), consider adding a minimum-bar persistence rule (e.g. three consecutive bars in the same cluster) to filter noise.

3. Prerequisites

Requirement	Minimum Version / Note
cTrader / cAlgo	v4.8 or later
.NET Core / .NET 6.0	Bundled in cAlgo
CPU	Quad-core $\geq$ 2.0 GHz
RAM	$\geq$ 8 GB
Internet connection	Required for licence validation / Telegram notifications
Telegram Bot Token	Generated via @BotFather

Tips

- Whitelist `api.clickalgo.com`:443 if behind a firewall.
- For multi-symbol, use cTrader workspaces or multiple charts.

4. Installation

Follow these familiar cTrader steps to install the indicator:

1. **Obtain the file**: download `K-Means Clustering (ML-Based).algo` from ClickAlgo.
2. **Compile**: select `K-Means Clustering (ML-Based)` in the list and click **Build**.
3. **Attach to chart**: drag the indicator from the list onto your chart.

> *Pro tip: if start-up feels slow, temporarily lower `TrainingHistoryBars` to 5 000 for a faster initial load.*

5. Licence Activation

On first load, the indicator validates your licence. If none is found:

1. Enter your email and activation key in the dialog.
2. Click *Activate*.

6. Project Architecture

This section outlines the core technical strengths and module responsibilities of the indicator, highlighting its performance, scalability, and maintainability.

High-Level Components

1. `K-Means Clustering (ML-Based)` (main class)
   - Orchestrates parameter handling, feature extraction, model pipeline, UI rendering, and alert routing.
   - Designed for thread-safe operation: all shared state (e. g. centroid statistics, label maps) is protected by lightweight locks.
2. `DataPoint` & `TransformedData` (DTOs)
   - `DataPoint` holds raw feature values for each bar.
   - `TransformedData` extends `DataPoint` with PCA feature vector and predicted label—enabling zero-allocation pipelines.
3. ML pipeline builder (`BuildPipeline`)
   - Composes a single IEstimator: Min–Max normalization → PCA projection → K-Means clustering.
   - Uses ML.NET's optimized routines with `KMeans++` initialization and multi-threaded solver.
4. Runtime services
   - **Training window manager**: maintains a rolling in-memory buffer of the last `HistoryBars` bars, sliding it on each tick.
   - **Retrain scheduler**: offloads model retraining to a background task, atomically swapping in the new MLContext and prediction engine without blocking the UI thread.
5. UI layer
   - Dynamic dashboard and legend built with cAlgo API `Canvas`/`ScrollViewer`, updated at a configurable refresh rate.
   - Custom bar-painting and text overlays avoid full redraws, achieving ∼40 bars/sec on modern CPUs.
6. Integrations
   - **LicenseChecker**: enforces versioned licensing with fail-safe postponement of initialization.
   - **TelegramMessenger**: synchronous HTTP calls wrapped in try/catch, with JSON validation to ensure reliable alert delivery.

Technical Strengths

- **Asynchronous Retraining**: Model refreshes run in parallel to tick processing, ensuring zero latency impact on live predictions.
- **Memory Efficiency**: Uses a pre-allocated pool of `DataPoint` objects to eliminate per-tick allocations and reduce GC pressure.
- **Modular Design**: Clear separation of concerns—feature computation, ML pipeline, UI rendering and alert logic—facilitates unit testing and future extensions.
- **Robust Thread Safety**: Critical sections are minimized to tiny lock scopes, preventing contention while preserving data integrity.
- **Configurable Performance**: All key parameters (window size, retrain interval, PCA components, iteration limits) are exposed, enabling tuning for any hardware profile.

- **Zero-Downtime Updates**: Atomic swap of the prediction engine and label-mapping ensures that no bars are skipped or misclassified during retraining.

Threading & Performance

A lightweight pool of `DataPoint` structs reduces GC pressure, while retraining runs off-thread and swaps the model atomically, ensuring no ticks are lost during execution.

7. User Interface Overview

Figure 1: High-level interface layout.

- **Cluster colouring**: bars coloured by regime.
- **Live labels**: ↑ bullish, ↓ bearish, — range.
- **Dashboard metrics**: sample counts, DB index, silhouette, cluster sizes.
- **Legend**: colour, average pip change, regime.

8. Detailed Parameters

8.1. K-Means Model Settings

Parameter	Description	Range	Default
Number of clusters (K)	Total clusters to learn	2–6	3
Training history bars	Bars used per training session	1 000–25 000	10 000
Validation data (%)	Percentage reserved for validation	1–50%	20%
Retraining interval (bars)	Retrain after this many new bars	1–100 000	5
PCA components	Principal components retained	1–22	1
Max K-Means iterations	Maximum solver iterations	1–20 000	100
Optimisation tolerance	Convergence threshold	1e-10–1e-3	1e-5

Recommended profiles

- *Scalping*: $K = 2$, history=5 000, PCA=1.
- *Intraday swing*: $K = 3$, history=12 000, PCA=3.
- *Higher time-frames (H1+)*: $K = 3$, history=20 000, retrain=10.

8.2. Feature Indicators (22 inputs)

Indicator	Defaults	Purpose
RSI	Period=14	Momentum extremes
ADX / ADXR / DI^+	Period=14	Trend strength
ATR	Period=14, EMA	Volatility measurement
Bollinger Bands	Period=20, StdDev=2, EMA	Volatility envelopes
MACD histogram	12 / 26 / 9	Trend momentum
Stochastic oscillator	14 / 3 / 3, EMA	Overbought / oversold cycles
EMA fast / slow	14 / 120	Trend crossover
CCI	Period=20	Price deviation
ROC	Period=12	Rate of change
VWAP (short / long)	14 / 28	VWAP deviation
OBV / MFI / SMA (vol.)	Defaults	Volume flow / smoothing

Tip: increase ATR period in low-volatility markets.

8.3. Telegram Alerts (optional)

Parameter	Description
Enable Telegram alerts	Toggle notifications on / off
Bot token	Token from `@BotFather`
Chat ID	Destination chat / group
Alert mode	{AllClusters, OnlyBullish, OnlyBearish, Off}

Setup workflow

1. Run `/newbot` in `@BotFather` and copy the token.
2. Send a message to your bot and retrieve the chat ID (e.g. via `@RawDataBot`).
3. Paste both values into the indicator parameters.
4. Select an alert mode and verify “

8.4. Cluster Colour Palette

Define up to six colours:

- $K = 2$: Colour1 bullish, Colour2 bearish.
- $K = 3$: Colour1 bullish, Colour2 bearish, Colour3 range.
- $K > 3$: Top three bullish $\rightarrow$ Colour1–3; top three bearish $\rightarrow$ Colour4–6.

Use named colours (e.g. `DodgerBlue`) or hex codes.

9. Mathematical Appendix

This appendix formalises the mathematics behind the indicator: (1) the exact definition of every input feature, (2) a concise derivation of Principal Component Analysis (PCA) and K-Means as used here, and (3) guidelines for choosing the optimal number of clusters K.

9.1. Exact Feature Definitions (22 Inputs)

#	Name	Formula
1	Log return	$r_t = \ln(\frac{C_t}{C_{t-1}})$
2	Candle body	$B_t = C_t - O_t$
3	Upper wick	$U_t = H_t - \max\{O_t, C_t\}$
4	Lower wick	$\ell_t = \min\{O_t, C_t\} - L_t$
5	RSI	$\mathrm{RSI}_t^{(p)}$
6	ATR	$\mathrm{ATR}_t^{(p)}$
7	ADX	$\mathrm{ADX}_t^{(p)}$
8	ADXR	$\mathrm{ADXR}_t^{(p)} = \frac{1}{2}(\mathrm{ADX}_t + \mathrm{ADX}_{t-p})$
9	DI^+	$DI_t^+ = 100\,\dfrac{\text{Smoothed } DM^+}{\mathrm{ATR}_t}$
10	OBV	$\mathrm{OBV}_t = \mathrm{OBV}_{t-1} + V_t\,(C_t - C_{t-1})$
11	MFI	$\mathrm{MFI}_t^{(p)}$
12	MACD hist.	$\mathrm{MACD}_{\text{fast,slow}} - \mathrm{DEA}_{\text{signal}}$
13	Stoch. %K	$100\,\dfrac{C_t - L_{p,t}}{H_{p,t} - L_{p,t}}$
14	Stoch. %D	$\mathrm{SMA}(\%K, p_D)$
15	BB%B	$\dfrac{C_t - \mathrm{BB}_{\text{low}}}{\mathrm{BB}_{\text{high}} - \mathrm{BB}_{\text{low}}}$
16	$\mathrm{EMA}_{\text{fast}}$	$\mathrm{EMA}(C_t, p_{\text{fast}})$
17	$\mathrm{EMA}_{\text{slow}}$	$\mathrm{EMA}(C_t, p_{\text{slow}})$
18	EMA diff	$\mathrm{EMA}_{\text{fast}} - \mathrm{EMA}_{\text{slow}}$
19	CCI	$\mathrm{CCI}_t^{(p)}$
20	ROC	$100\,\dfrac{C_t - C_{t-p}}{C_{t-p}}$
21	VWAP deviation	$C_t - \mathrm{VWAP}_t^{(p)}$
22	Volume ratio	$\dfrac{V_t}{\mathrm{SMA}(V, p)}$

Table 1: Definitions of the 22 input features (period p, prices O_t, H_t, L_t, C_t, volume V_t).

9.2. Principal Component Analysis (PCA)

Let the zero-mean feature matrix be $\mathbf{X} \in \mathbb{R}^{n \times d}$. The empirical covariance is $\mathbf{\Sigma} = \frac{1}{n-1}\mathbf{X}^\top \mathbf{X}$. Eigen-decomposition

$$\mathbf{\Sigma} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top, \qquad \mathbf{\Lambda} = \operatorname{diag}(\lambda_1 \geq \cdots \geq \lambda_d)$$

provides orthonormal eigenvectors $\mathbf{Q} = [\mathbf{q}_1, \ldots, \mathbf{q}_d]$. Retaining the first m eigenvectors ($m =$ `PcaComponents`) yields the projection

$$\mathbf{Z} = \mathbf{X}\mathbf{Q}_m \in \mathbb{R}^{n \times m},$$

maximising retained variance $\sum_{i=1}^{m} \lambda_i$.

9.3. Mini-Batch K-Means

Given $\mathbf{Z}$ the algorithm minimises

$$\mathcal{L} = \sum_{i=1}^{n} \left\| \mathbf{z}_i - \boldsymbol{\mu}_{\kappa(i)} \right\|^2, \qquad \kappa(i) = \arg\min_{k \in \{1,\ldots,K\}} \|\mathbf{z}_i - \boldsymbol{\mu}_k\|,$$

by alternating E-steps (cluster assignment) and M-steps (centroid update). We employ `KMeans++` initialisation and stop when $\|\boldsymbol{\mu}^{(t)} - \boldsymbol{\mu}^{(t-1)}\|_\infty < \varepsilon$ or after `MaxIterations`.

9.4. Choosing the Number of Clusters K

- **Elbow rule**: plot total within-cluster SS vs. K and pick the elbow where marginal gain flattens.
- **Silhouette maximisation**: choose K that maximises mean silhouette S. In practice $S > 0.5$ is sought.
- **Davies–Bouldin minimisation**: select the lowest DB index; values ≤ 1 indicate good separation.
- **Domain prior**: for intraday FX, $K \in \{2, 3\}$ captures {trend up, trend down, range}; higher K adds granularity but raises over-fitting risk.

10. Practical Strategy Examples

The K-Means regime label acts as a *context switch*: when the market is in a directional cluster you deploy trend-following tools, and when the label signals a sideways cluster you pivot to mean-reversion logic. The two "playbooks" below are written in plain language (no code) so you can translate them to any platform.

10.1. EMA Trend-Following Playbook

When to activate: current cluster $\in \{\texttt{BullTrend}, \texttt{BearTrend}\}$, i.e. high average pip-delta and strong ADX projection.

1. **Trend confirmation:** wait for a dual-EMA crossover.
   - Use EMA_{50} vs. EMA_{200} for H1–D1 charts; EMA_{14} vs. EMA_{120} for M5–M30.
   - A "golden cross" (fast EMA closing above slow EMA) biases longs; a "death cross" biases shorts.[1]
2. **Entry:** open in the direction of the crossover *only* if the cluster label still reads `BullTrend` or `BearTrend`.
3. **Stop-loss:** place at $0.7 \times \text{ATR}_{14}$ beyond the slow EMA; this keeps the stop outside normal volatility.[2]
4. **Exit:**
   - Opposite EMA crossover *or*
   - Cluster flips to a non-trend label.
5. **Position size:** risk 1 % of account equity per trade ("fixed fractional" rule).[3]

Why it works. Moving averages excel at capturing sustained bias but suffer in chop.[4, 5] The regime filter blocks most of that chop, improving win rate without changing the EMA logic.

10.2. Range Oscillator Playbook

When to activate: current cluster = `Range`, i.e. low directional drift, high inside-bar frequency.

1. **Overbought/oversold test:**
   - Price touches Bollinger lower band *and* Stochastic $< 20 \Rightarrow$ set up a long.
   - Price touches Bollinger upper band *and* Stochastic $> 80 \Rightarrow$ set up a short.
   - Ignore signals if band width $< 0.3\%$ of price (volatility too compressed).[4]
2. **Entry trigger:** confirmation candle closing back inside the bands.
3. **Stop-loss:** $0.5 \times \text{ATR}_{14}$ beyond the entry candle's extreme.
4. **Profit target:** ATR_{14} or opposite Bollinger band, whichever comes first.
5. **Cooldown:** stand aside for three bars after a cluster change or after a full target hit (prevents over-trading).

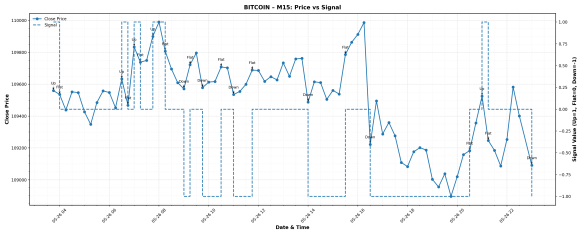
Why it works. The stochastic oscillator is explicitly bounded [0,100], making it ideal for range contexts.[8, 6, 9] Coupling it with Bollinger bands aligns momentum exhaustion with statistically unlikely price extremes, while the K-Means label keeps you out of nascent breakouts.

[1] See the golden-cross definition in [1].
[2] ATR-based placement reduces premature stop-outs [2].
[3] Fixed-risk sizing is a staple of professional money management [3].
[4] Band-width filters avoid signals during volatility droughts [7].

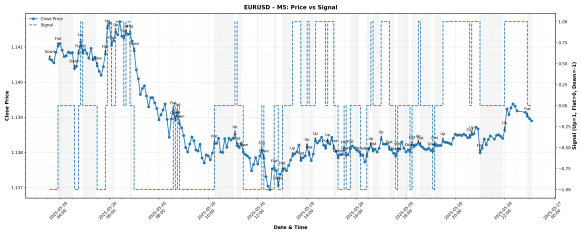
11. Backtesting

This section details the predictive quality obtained from the warning messages sent by Telegram, avoiding evaluating validation data that includes the new candles generated after each retraining of the model. In this way, the results reflect real-time data or the approximate performance of a day when checking market regimes.

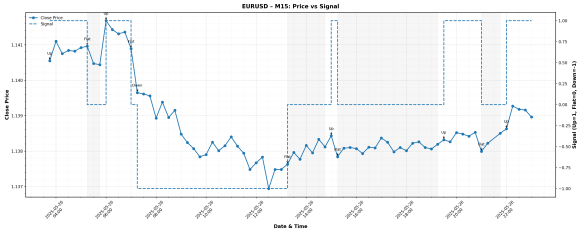
Figure 2: Full indicator panel showing parameters, validation dashboard and colour-coded bars.



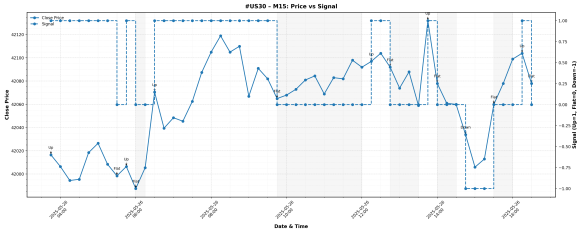
(a) Bitcoin M15: volatile regime shifts.



(b) EURUSD M5: frequent short-term clustering.



(c) EURUSD M15: extended range phases.



(d) US30 M15: clear regime detection.

Figure 3: Real-time regime classification examples across instruments and time-frames.

In the context of this indicator, the following table shows the best-found hyperparameters per symbol and timeframe, based on balanced accuracy, F1–macro score, Cohen's kappa and regime hit rate over a one-day backtest:

Symbol	TF	K	History	Balanced ACC	F1 Macro	Kappa	Hit Rate
EURUSD	M5	0.250	4	0.448	0.206	0.000	0.500
EURUSD	M15	0.250	13	0.490	0.219	0.000	0.500
BITCOIN	M15	0.250	1	0.547	0.236	0.000	0.526
#US30	M15	0.281	8	0.627	0.481	0.353	0.444

Table 2: Best hyperparameters per symbol/timeframe from one-day backtest.

Recommended configurations Based on these results:

- **EURUSD M5**: Use $K = 0.25$, history window of 4 bars for faster adaptation in short-term scalping.
- **EURUSD M15**: $K = 0.25$, history of 13 bars balances accuracy and stability on 15-minute charts.
- **Bitcoin M15**: $K = 0.25$, minimal history (1 bar) captures quick regime shifts in volatile crypto.
- **#US30 M15**: $K = 0.281$, history of 8 bars maximises regime detection in index trading.

What is K? In this study K is **the sensitivity threshold** applied to the regime-change test: the panel issues a new colour (Uptrend, Downtrend or Range) when the close–to–close movement of a candle exceeds K times the recent volatility (ATR_{14} on the same chart). A *smaller* K (e.g. 0.20) triggers earlier but noisier signals, whereas a *larger* K (e.g. 0.35) waits for a more pronounced move and filters out micro-noise. Note that this K is **independent of the number of clusters used by the K-Means engine**; it strictly controls how reactive the visual regime detector is.

On the role of K and its equivalence to a triple-barrier scheme In our one-day backtest, the factor K is *not* offered as a live, user-tunable input, but is instead the calibrated multiplier that defines the upper and lower price barriers—analogous to the profit-taking and stop-loss levels in a standard *triple-barrier* labeling process. Concretely, for each symbol and timeframe we place two horizontal barriers at

$$\text{Upper barrier} = P_{\text{close}} + K \cdot \text{ATR}_{14}, \quad \text{Lower barrier} = P_{\text{close}} - K \cdot \text{ATR}_{14},$$

and a third "time barrier" of one candle. We then classify the regime by which barrier is hit first, or by time expiry (range). The K values in Table 2 are those that, under these equivalent triple-barrier conditions, maximize balanced accuracy, F1–macro, Cohen's kappa and regime hit rate over a full trading day.

11.1. User-friendly presentation of results

To let traders verify that the panel *really* detects regime changes—and to show where the numbers in Table 2 come from—we follow the procedure summarized in Table 3. This design avoids any "peeking into the future" by using only the alerts the user actually receives via Telegram.

Step	What we did	Why it matters
1	**A typical trading day**. We ran a full session (Europe + US) per symbol and timeframe.	Exactly replicates what the trader sees live; no "future" data.
2	Only the **Telegram alerts** were used. Each time the panel changes color, the message is logged verbatim.	Evaluates the indicator's *real* output, not a retroactive fit.
3	We **excluded** the candles used for model retraining (*live retrain*).	Prevents "cheating" by scoring on data the model has already seen.
4	Each signal is compared to the **next candle**. If it says *Uptrend* and that candle closes higher → hit; if it says *Range* and price stays within $\pm\frac{1}{2}$ ATR → hit.	An objective, easy-to-verify rule.

Table 3: Evaluation methodology without forward-looking data.

What the table columns measure

Balanced ACC "How balanced are the hits across up, down, and range?" 1.00 = perfect; 0.33 random guess among three options.

F1 Macro "How many good signals versus false ones, averaged over the three classes?"

Cohen's Kappa "How much better than pure chance?" 0 = random; 1 = perfect agreement.

Hit Rate "Raw percentage of correct signals," ignoring class imbalance.

How to read Table 2 without overload

- Blocks are grouped by instrument, e.g. **EURUSD (M5 and M15)**.
- The recommended row is shown in **bold** or marked with ✓.
- Icons match the panel: bullish, bearish, ■ range.
- Metric cells are shaded from red yellow green for instant readability.
- Sample footer text:
  "With this setup the panel hits ∼ 50% of regime changes and, crucially, shows no bias to one side (Balanced ACC 0.49)."

What this back-test *does not* measure

- Distance to take-profit or stop-loss.
- Trade management or risk path.
- Exact magnitude of the subsequent move.

That is why some pairs show Kappa = 0: they get direction right only half the time, though with modest statistical edge. The goal is **not** to time perfect entries but to **signal regime change**.

Summary for the reader

1. Fixed day, real alerts, next-candle comparison, exclusion of retrain.
2. Each metric translated into a practical phrase.
3. Colors and symbols make the table intuitive.
4. Transparency about what the numbers omit.

This way, users see *real*, understandable, and honest results about the indicator's capabilities.

12. Execution Flow

1. `Initialise()`: licence validation, then `HeavyInit()`.
2. `HeavyInit()`: timeframe / market checks, history load, indicator init, initial model train, UI build.
3. `Calculate(index)`:
   - Feature vector → predict cluster.
   - Colour bar, add live symbol.
   - Update stats, send alerts.
   - Retrain every `Retraining Interval` bars.

Performance: ~0.7/s total process on Intel Xeon e5-2697v3 with 10 000 history bars.

13. Colour & Symbol Interpretation

Colour	Symbol	Regime	Action
Blue (Colour1)	↑	Uptrend ($\Delta > 0$)	Enter longs on pullbacks; stop near EMA14
Yellow (Colour2)	↓	Downtrend ($\Delta < 0$)	Enter shorts on rallies; stop near EMA14
Grey (Colour3)	–	Range ($\Delta \approx 0$)	Stand aside or trade breakouts
Colours 4–6	↑/↓	Secondary clusters	Fine-tune entries / exits

14. Best Practices

- Maintain validation $\geq 15\%$.
- Use consistent parameters across correlated symbols.
- Increase the retraining interval when running multiple instances.
- Back-test cluster performance before live trading.
- Use OnlyBullish / OnlyBearish for targeted alerts.

15. Troubleshooting

Symptom	Likely Cause	Solution
Unsupported timeframe	Outside M1–D1	Switch timeframe to M1–D1
Insufficient data	Too few bars	Reduce history or load more data
ObjectDisposedException	Retraining in progress	Increase retraining interval
Telegram HTTP/API error	Invalid credentials	Verify bot token / chat ID and network connectivity
Unstable metrics	Small history or extreme validation %	Increase history or adjust validation %

16. FAQ

More than 25 000 bars?
Yes—adjust the code limit, but expect higher memory/time.

Works 24/7 crypto?
Yes, if `IsMarketOpen()` is disabled or handles 24/7.

Persist models between sessions?
Not currently; each session retrains fresh.

Export centroids?
Not directly; export `_trainSet` and predictions.

Disable a feature?
Comment out its init in `InitIndicators()`.

17. Version History

- 1.1.0 (2025-05-15): DataPointPool, Telegram modes, GC optimisations, colour logic ,Major refactor — new regime engine, breaking parameter names, UI improvements.
- 1.0.0 (2023-11-01): Initial release.

References

[1] Investopedia, “Golden Cross Pattern—Definition Examples,” accessed May 2025.

[2] EarnForex, “Position Sizing Rules,” 2024.

[3] Reddit /r/Daytrading, “The Importance of the Fixed Risk Rule,” 2023.

[4] TrendSpider, “Moving Average Crossover Strategies,” 2024.

[5] Investopedia, “How to Use a Moving Average to Buy Stocks,” 2024.

[6] Oanda, “Mastering Stochastic Oscillator Trading Strategies,” 2024.

[7] TradingView Blog, “Bollinger Bands Basics and Breakout Strategy,” 2025.

[8] Investopedia, “Stochastic Oscillator: What It Is and How It Works,” 2024.

[9] Investopedia, “Keltner vs Stochastic for Volatile Markets,” 2025.