KruegerAlgorithms
Quantitative Marktdaten-Analyse

Trading-Strategie mit KI erstellen: die Fehler, die im Code nicht zu sehen sind

Ein Sprachmodell schreibt in Minuten einen Expert Advisor, der kompiliert und im Backtest gewinnt. Die Fehler stecken selten in der Syntax. Sie stecken in dem, was der Code über den Markt annimmt: Information, die zum Einstieg noch nicht da war, Fills, die es nie gab, Kosten, die fehlen, Handelszeiten, die stimmen, bis die Uhr umgestellt wird. Zwölf Fehlerklassen, jede mit der Zahl aus unseren eigenen Tests oder aus einem Kundenfall, und die Prüfliste dagegen.

Dokument-Metadaten

Feld	Wert
Herausgeber	KruegerAlgorithms
Titel	Trading-Strategie mit KI erstellen: die Fehler, die im Code nicht zu sehen sind
Erscheinungsdatum	18. September 2026
Version	v1.0
Verdikt	Referenz
Datengrundlage	**Grundlage:** Dies ist keine Messung, sondern ein Nachschlagetext. Jede Zahl darin stammt aus einer verlinkten Studie dieser Seite, aus einem dokumentierten Fall der Auftragsarbeit (anonymisiert) oder ist als interne, noch unveröffentlichte Messung gekennzeichnet. Die Fehlerklassen sind nicht KI-spezifisch. Wir haben jede davon selbst gebaut, mit Jahren Markterfahrung und einem Prüf-Workflow. Der Unterschied ist die Geschwindigkeit, mit der ein Modell sie baut, und dass dabei niemand hinschaut. **Keine Handelsempfehlung.**
Hinweise	Hinweis vom 29. September 2026; Neurechnung vom 5. Oktober 2026. Wortlaut in der Zusammenfassung.
Dokumenttyp	Statistische Auswertung (keine Trading-Empfehlung)

Rein statistische Auswertung historischer Marktdaten. Keine Trading-Empfehlung, keine Finanzberatung. Historische Performance ist keine Garantie für die Zukunft. Trading mit Hebelprodukten kann zum Totalverlust führen.

Zusammenfassung

Hinweis vom 29. September 2026: Ein Teil der R-Werte dieser Studie stammt aus unserer älteren Backtest-Kette, die auch auf einem Zufallsmarkt ohne Richtung positive R bucht. Diese Werte sind zu hoch; Ergebnisse, die nicht an der Kette hängen (Preisstatistik, Zählungen), sind nicht betroffen. Details: Der Zufallsmarkt als Lineal.

Neurechnung vom 5. Oktober 2026: Die 25 Kombinationen aus Ausbruchsregel und Markt aus unserem Regelwerk sind auf der korrigierten Kette und auf echten Broker-Ticks neu gemessen. Keine hat einen Vorteil gegenüber dem Zufallsmarkt, im Mittel bleiben nach Kosten −0,08 R je Trade. Die betroffenen R-Werte dieser Studie sind damit nicht belegt, die übrigen Ergebnisse bleiben unberührt. Alle Zahlen: Unsere eigenen Ausbruchsregeln, neu gemessen.

Ein Sprachmodell schreibt in Minuten einen Expert Advisor, der kompiliert und im Backtest gewinnt. Die Fehler stecken selten in der Syntax. Sie stecken in dem, was der Code über den Markt annimmt: Information, die zum Einstieg noch nicht da war, Fills, die es nie gab, Kosten, die fehlen, Handelszeiten, die stimmen, bis die Uhr umgestellt wird. Zwölf Fehlerklassen, jede mit der Zahl aus unseren eigenen Tests oder aus einem Kundenfall, und die Prüfliste dagegen.

Die Szene ist inzwischen Alltag: Eine Regel im Kopf, ein Prompt an ChatGPT oder Claude, drei Minuten später ein Expert Advisor für MetaTrader 5 oder eine Pine-Strategie für TradingView. Der Code kompiliert. Der Backtest ist grün. Was fehlt, ist nichts, was man im Code sehen könnte. Es fehlt das, was der Markt weiß und das Modell nicht: wann eine Information existiert, was ein Fill kostet, wie ein Broker zählt, wann eine Session wirklich beginnt.

Dieser Text sortiert die Fehler, die dabei entstehen, in drei Gruppen: der Backtest lügt, der Code stimmt aber der Markt ist anders, die Idee trägt nicht. Am Ende steht die Prüfliste, mit der man sie selbst findet. Sie ist dieselbe, die hier vor jeder Auftragsarbeit läuft.

Diese Auswertung enthält keine Trading-Empfehlung.

Inhaltsverzeichnis

Zusammenfassung **2**

Inhaltsverzeichnis **3**

1. Worauf das Modell optimiert **4**

2. Der Backtest lügt **4**

2.1 Information aus der Zukunft . . . 4

2.2 Der Trigger kommt spät, der Einstieg bleibt früh . . . 4

2.3 Fills, die es nie gab . . . 5

2.4 Kosten, die fehlen oder pauschal sind . . . 5

2.5 Trefferquote ohne Maßstab . . . 5

2.6 Zwei Jahre sind ein Regime, kein Beweis . . . 5

3. Der Code stimmt, der Markt ist anders **6**

3.1 Handelszeiten und Sommerzeit . . . 6

3.2 Punkte, Pips, Tickwert, Symbol . . . 6

3.3 Zustand statt Ereignis . . . 6

3.4 Brutto und netto . . . 6

3.5 Die vier Lebenszyklen . . . 7

4. Die Idee trägt nicht, und der Code kann es nicht wissen **7**

5. Die Prüfliste **7**

6. Der Code ist der billige Teil **8**

Vorbehalt **8**

1. Worauf das Modell optimiert

Ein Sprachmodell kann Syntax, Struktur und die Dokumentation einer Plattform. Es schreibt Boilerplate schneller als jeder Mensch, kennt die Signaturen von `OrderSend` und `strategy.entry` und macht dabei weniger Tippfehler als wir. Das ist echter Wert, und wir nutzen ihn auch.

Es optimiert aber auf etwas anderes als auf den Markt: auf die Zufriedenheit der Person im Chat. „Mach den Backtest profitabel“ ist eine Anweisung, die ein Modell auf drei Wegen erfüllen kann, und alle drei sehen im Chat wie Erfolg aus. Es kann Information einbauen, die zum Zeitpunkt des Einstiegs noch nicht existierte. Es kann Parameter so lange drehen, bis die Kurve passt. Es kann den Simulator so bauen, dass Fills entstehen, die es real nie gäbe. Keiner dieser Wege ist Betrug. Es sind die drei natürlichen Abkürzungen, die jeder nimmt, der einen Backtest ohne Gegenprobe baut, und ein Modell nimmt sie ohne das Unbehagen, das ein erfahrener Entwickler dabei spürt.

Markterfahrung heißt an dieser Stelle nicht, den Markt besser vorherzusagen. Es heißt, die Liste der Dinge zu kennen, die in keiner API-Dokumentation stehen: dass die 30-Minuten-Opening-Range um 09:05 Uhr noch nicht existiert, dass der Spread im DAX um 07:38 Uhr ein anderer ist als um 10:00 Uhr, dass zwei Broker denselben Index mit unterschiedlicher Punkteskala führen, dass Europa und die USA an verschiedenen Wochenenden die Uhr umstellen. Jeder Punkt unten ist so ein Eintrag auf dieser Liste.

2. Der Backtest lügt

2.1 Information aus der Zukunft

Lookahead ist der Fehler, vor dem jeder warnt und den trotzdem jeder baut. Nie plump, nie mit dem Schlusskurs von morgen. Sondern mit einer Größe, die *für den Tag* definiert ist und an einer Stelle abgefragt wird, an der der Tag noch nicht so weit ist.

Uns ist das mit vier Kontext-Features passiert, trotz Prüfroutine: Die Grenzen der 30-Minuten-Opening-Range wurden für Einstiege innerhalb der ersten 30 Minuten gelesen, eine Richtungs-Tendenz relativ zum Cash-Open für Einstiege vor dem Cash-Open. Betroffen waren 60.129 von 123.193 Episoden. Der Prognose-Spread eines darauf trainierten Modells fiel nach dem Fix von +0,726 R auf +0,191 R. Die volle Diagnose steht im Post-mortem.

Im generierten Code sieht das so aus: `iHigh(_Symbol, PERIOD_D1, 0)` als „Tageshoch“ um 10 Uhr. Eine Opening-Range, die ab der ersten Minute als Filter dient. In Pine ein `request.security` auf den höheren Timeframe ohne `[1]`, das den laufenden Tagesbalken liest. Und die ganze Familie der zentrierten Indikatoren: TMA-Bänder, ZigZag, zentrierte Durchschnitte, alles, was für eine Kerze Werte aus den Kerzen *danach* mittelt.

Ein Kundenfall vom September 2026: ein erkennbar KI-generiertes Skelett, das per `iCustom` TMA-Bänder mit `HalfLength 32` auf H4 aufruft. Der Chart zeigte drei Pfeile in sieben Wochen, alle sauber an den Bandkreuzungen. Nur: Die Bänder werden für vergangene Kerzen nachträglich neu gezeichnet, die Pfeile sind Rückschau. Die Variante ohne Repaint läuft bei dieser Einstellung über fünf Handelstage hinterher. Der Kunde hätte einen EA bekommen, der seinen Chart nie reproduziert, und die Reklamation wäre nach dem ersten Live-Vergleich gekommen.

Prüfung: Für jede Eingangsgröße den Zeitpunkt notieren, ab dem sie bekannt ist, und mit dem Einstiegszeitpunkt vergleichen. Diagnose-Signatur: Der „Edge“ konzentriert sich in den Trades, bei denen das Feature noch unbekannt war.

2.2 Der Trigger kommt spät, der Einstieg bleibt früh

Die zweite Klasse ist subtiler. Ein Filter wartet auf eine Bestätigung, zum Beispiel Volumen. Die Bedingung wird erst später wahr, der Einstiegspreis bleibt aber rechnerisch am ursprünglichen Level. Der Backtest kauft dann zu einem Preis, den es zum Zeitpunkt der Entscheidung längst nicht mehr gab.

Genau das hat uns ein Setup in der US-Mittagsflaute beschert: +0,90 R bei 60 % Trefferquote mit Volumen-Gate, das beste Setup des ganzen Systems. Ohne das Gate: −0,11 R bei 29 %. Das Gate hatte den Trigger im Median um 100 Minuten verschoben, während der Einstieg am alten Level blieb. Der komplette Edge war Lookahead in Verkleidung. Die Studie dazu: Phantom-Edges durch Volumen-Gate.

Prüfung: Den Abstand zwischen „Level erreicht“ und „Bedingung wahr“ loggen (Median über 10 Minuten ist Alarm), und den Abstand zwischen Fill-Preis und Level in R (Median über null heißt fiktiver Einstieg).

2.3 Fills, die es nie gab

Ein Backtester auf Kerzenbasis muss raten, was innerhalb einer Kerze passiert ist. Liegen Stop und Ziel in derselben Kerze: Was wurde zuerst berührt? Wird ein Limit am Level gefüllt oder am Open der nächsten Kerze? Ein generierter Simulator entscheidet das im Zweifel zugunsten der Strategie, nicht aus Absicht, sondern weil die optimistische Variante die einfachere ist.

Unser Beleg: Ein Doji-Muster auf dem 2-Minuten-Chart zeigte auf echten Daten +0,07 R netto mit einem t-Wert von +16. Ein synthetischer Random Walk, durch dieselbe Bracket-Kette geschickt, lieferte für dasselbe Muster +0,16 R brutto. Der Edge war eine Eigenschaft der Kette, nicht des Marktes. Nachzulesen unter Doji-Falle im 2-Minuten-Chart.

Prüfung: Dieselbe Engine auf einen Zufallspfad loslassen. Was sie dort „findet“, ist die Engine. Das ist der Random-Walk-Anker, und er ist bei Minuten-Brackets Pflicht. Und wo die Reihenfolge innerhalb einer Minute das Ergebnis entscheidet, hilft kein Raten, sondern nur die Tick-Reihe: In unserer DAX-Auswertung der Eröffnungskerze wurden an 250 Tagen beide Seiten in derselben Minute gebrochen. 223 davon ließen sich mit Tick-Daten auf Millisekunden-Basis auflösen, die übrigen 27 blieben auch auf Tick-Ebene unentschieden. Minutendaten sind die Basis, Ticks das Werkzeug für genau diese Fälle.

2.4 Kosten, die fehlen oder pauschal sind

Generierter Code rechnet ohne Spread, oder mit einem pauschalen Prozentsatz aus einem Aktien-Tutorial. Beides ist für Intraday-Trades wertlos. Die Kennzahl, die vor dem Backtest entscheidet, ist der Spread relativ zur Stop-Distanz, weil jeder Trade diese Toll vorab bezahlt.

Beim Test einer Index-Erweiterung sah ein Spread von 0,80 Punkten klein aus. Relativ zum Stop des Setups war er rund ein Viertel des Risikos je Trade, und der Markt war damit tot, bevor die Edge-Frage überhaupt gestellt war. Der Index mit dem kleinsten absoluten Spread der Liste war der teuerste Markt. Alles ab Euro Stoxx 50 lag beim Sieben- bis Neunfachen der DAX-Toll. Die Rechnung: SMI, CAC und CA60 im Test.

Prüfung: Kosten je Trade als Anteil von R ausweisen, und jede Kennzahl netto. Was nur brutto funktioniert, funktioniert nicht.

2.5 Trefferquote ohne Maßstab

Das Modell meldet „67 % Trefferquote“ als Erfolg. Bei einem Stop, der doppelt so weit weg ist wie das Ziel, sind 67 % exakt das, was ein Münzwurf liefert: Stop geteilt durch (Stop plus Ziel). Die Formel ist über 300 Jahre alt, und 35.000 Münzwurf-Trades je Markt auf DAX, NQ und SPX bestätigen sie auf ein bis zwei Prozentpunkte. Die Herleitung in einfacher Sprache: Der Zufalls-Maßstab.

Prüfung: Trefferquote minus Zufalls-Trefferquote, Trade für Trade, weil jeder Trade sein eigenes Stop-Ziel-Verhältnis hat.

2.6 Zwei Jahre sind ein Regime, kein Beweis

Der Standard-Backtest aus dem Chat läuft über die Daten, die gerade kostenlos verfügbar sind, meist ein bis zwei Jahre. Sechs Breakout-Setups über elf Jahre zeigen, warum das nicht reicht: 2015 bis 2017 brachten +15 R bei einem Drawdown von −449 R, die acht Jahre danach +621 R pro Jahr bei −136 R Drawdown. Dieselben Regeln,

zwei Verteilungen. Wer nur eine davon gesehen hat, weiß nicht, was ein schlechter Monat ist. Die Studie: Ein Breakout-Buch über elf Jahre.

Und selbst elf Jahre schützen nicht vor dem Verfall: In unserer Deployment-Studie blieb der Backtest grün, während das echte Konto ab 2025 verlor. Phase 5 ist das ehrlichste Kapitel der Serie, genau deshalb.

Prüfung: Alle Regimes, die die Daten hergeben. Walk-Forward statt einer Optimierung. Und eine Annahme darüber, wie schnell der Edge verfällt, bevor Geld dranhängt.

3. Der Code stimmt, der Markt ist anders

Die Fehler in diesem Abschnitt findet kein Backtest, weil der Backtest sie teilt. Sie tauchen erst auf dem Konto auf.

3.1 Handelszeiten und Sommerzeit

`if(Hour() == 15)` ist die häufigste Zeile in generierten Session-EAs. Sie stimmt, solange Serverzeit und Börsenzeit im gleichen Abstand bleiben. Europa und die USA stellen die Uhr aber an verschiedenen Wochenenden um. Zwei- bis dreimal im Jahr verschiebt sich das US-Open relativ zur Serverzeit um eine Stunde, für zwei bis drei Wochen. Ein EA, der das nicht kennt, handelt in diesen Wochen still eine andere Strategie. Dazu kommen die Kleinigkeiten, die kein Modell aus dem Prompt ableitet: Xetra schließt um 17:30 Uhr, der CFD läuft bis 22:00 Uhr; US-Halbtage; Feiertage, an denen der Kurs steht, der Spread aber nicht.

Prüfung: Session-Grenzen aus der Börsenzeitzone ableiten, nie aus der Serverstunde. Und einmal die Woche des DST-Versatzes im Tester anschauen.

3.2 Punkte, Pips, Tickwert, Symbol

Bei einem Broker ist ein Indexpunkt zehn Broker-Points, beim nächsten hundert. Derselbe EA mit denselben Eingaben ist damit auf dem zweiten Konto um den Faktor zehn daneben: Der Spread-Filter filtert nichts mehr, der Trigger verlangt zwei statt 0,2 Indexpunkte. Wir haben das beim Umzug zwischen zwei Brokern selbst durchgerechnet, jede punktbasierte Eingabe musste durch zehn. Dazu der Tickwert in Euro oder Dollar, und Symbolnamen, die sich zwischen `GER40`, `GER40.cash` und `DE40` unterscheiden. Generierter Code übernimmt „100 Points = 10 Pips“ aus einem Forex-Tutorial und rechnet damit auf Indizes.

Prüfung: `SYMBOL_POINT`, `SYMBOL_TRADE_TICK_VALUE` und die Kontowährung beim Start loggen, alle Distanzen in Indexpunkten oder in R denken, nie in Broker-Points.

3.3 Zustand statt Ereignis

Zurück zum Kundenfall aus 2.1. Das Skelett hatte, neben dem Repaint, vier weitere Fehler auf vielleicht 150 Zeilen. Die Kaufbedingung prüfte, ob das grüne Band *unter* dem weißen liegt, nicht, ob es *darunter gekreuzt* hat. Solange das Band unten blieb, hätte der EA nach jedem geschlossenen Trade sofort den nächsten eröffnet; im Chart war es ein Pfeil. Das Bar-Close-Gate wurde erst nach dem ersten Trade scharf, davor wertete der Code jeden Tick aus. Derselbe Indikatorname bekam einmal vier und einmal fünf Parameter; `iCustom` übergibt sie positional, mindestens ein Aufruf lieferte Unsinn. Und die H1-Bänder auf dem H4-Chart wurden mit `shift=1` gelesen, also die letzte H1-Kerze statt der H4-Kerze.

Keiner dieser Fehler ist exotisch. Es sind die Standardfehler, die jeder Anfänger einmal baut. Das Modell baut alle vier auf einmal und liefert dazu einen Kommentarblock mit „Copyright 2026“.

Prüfung: Jede Einstiegsbedingung als Ereignis formulieren (vorher falsch, jetzt wahr), das Gate vor dem ersten Trade beobachten, jeden Fremdindikator-Aufruf gegen seine Input-Liste halten.

3.4 Brutto und netto

Ein Kunden-EA sollte alle Positionen schließen, sobald der Zyklusgewinn ein Ziel erreicht. Er schloss zu früh: Die Summe zählte nur `DEAL_PROFIT`, ohne Swap und Kommission. Brutto stand das Ziel, netto nicht. Nach dem Fix fand der Kunde dieselbe Falle eine Ebene tiefer, in der Recovery-Logik, weil wir den Bug als Fundstelle statt als Klasse behoben hatten. `DEAL_PROFIT` ist das Feld, das das Modell nimmt, weil es das offensichtliche ist.

Prüfung: Jede Stelle, an der Geld summiert wird, bewusst entscheiden: Swap und Kommission dazu oder nicht. Und jeden gefundenen Bug als Klasse durch den ganzen Code suchen.

3.5 Die vier Lebenszyklen

Terminal-Neustart, Chart schließen, EA entfernen, Parameter ändern: In jedem dieser vier Fälle stirbt der Zustand im Speicher. Was bleibt, was resettet? Generierter Code hält Zähler in statischen Variablen und ist nach dem ersten Neustart ein anderer EA. Dazu ein Fall aus dem MQL5-Forum vom September 2026: Ein EA lief im Tester korrekt und auf dem Prop-Konto falsch, weil der Server dort Deal-Kommentare umschreibt oder leert und der EA seine Trades über den Kommentar wiedererkannt hatte.

Prüfung: Alle vier Lebenszyklen einmal auslösen und den Tick danach beobachten. Trades über Magic Number und Ticket identifizieren, nie über den Kommentar.

4. Die Idee trägt nicht, und der Code kann es nicht wissen

Die dritte Gruppe hat nichts mit Programmierung zu tun. Das Modell baut, was der Prompt verlangt, und wenn der Prompt „elf Indikatoren, Pfeil nur bei Voll-Konsens“ verlangt, baut es das sauber. Wir haben so einen Konsens-Algorithmus gemessen (interne Messung vom 18.09.2026, noch nicht veröffentlicht): Die elf Stimmen aus derselben Schlusskurs-Reihe sind im Mittel mit 0,41 korreliert, effektiv sind es 3,3 unabhängige Stimmen. Der Konsens kommt im Median 16 Kerzen nach dem ersten Kipper, der Kurs ist bis dahin 2,7 ATR in Signalrichtung gelaufen. Netto war das Ergebnis in allen 15 getesteten Markt-Timeframe-Zellen negativ.

Die zweite Falle dieser Gruppe ist die Auswahl. Wer das Modell zwanzig Varianten bauen lässt und die beste nimmt, hat mit hoher Wahrscheinlichkeit Rauschen ausgewählt. Bei zwanzig Versuchen ist ein t-Wert über 2 zu erwarten, ohne dass irgendetwas dahintersteht. Der Maßstab ist dann nicht „t gegen null“, sondern „wie stark ist der beste von zwanzig Zufallsläufen“.

Prüfung: Vor dem Bau die Frage stellen, welche Information die Regel enthält, die nicht schon im Kurs steckt. Und die Anzahl der probierten Varianten mitzählen.

5. Die Prüfliste

Zwölf Fragen, jede mit Ja oder Nein zu beantworten, bevor Geld an einer Regel hängt. Es ist dieselbe Liste, die hier vor jeder Auftragsarbeit läuft.

#	Frage	Wo es schiefgeht
1	Ist für jede Eingangsgröße notiert, ab wann sie bekannt ist?	2.1
2	Repaintet ein Indikator (zentriert, HalfLength, ZigZag, laufender HTF-Balken)?	2.1
3	Liegt der Median-Abstand zwischen Level und Trigger unter 10 Minuten, der Fill am Level?	2.2
4	Findet die Engine auf einem Random Walk einen Edge?	2.3
5	Stehen Kosten als Anteil von R je Trade da, und ist jede Kennzahl netto?	2.4
6	Ist die Trefferquote gegen Stop / (Stop + Ziel) gerechnet?	2.5
7	Umfasst der Test mehrere Regimes, mit Walk-Forward statt einer Optimierung?	2.6
8	Kommen Session-Grenzen aus der Börsenzeitzone, nicht aus der Serverstunde?	3.1

#	Frage	Wo es schiefgeht
9	Sind Point, Tickwert und Kontowährung geloggt, alle Distanzen in Indexpunkten oder R?	3.2
10	Ist jede Einstiegsbedingung ein Ereignis, und wurde jedes Gate vor dem ersten Trade beobachtet?	3.3
11	Ist an jeder Geldsumme entschieden, ob Swap und Kommission dazugehören?	3.4
12	Wurden Neustart, Chart zu, EA entfernen und Parameter ändern einmal ausgelöst?	3.5

Wer alle zwölf mit Ja beantworten kann, hat ein System, das den Backtest verdient, den es zeigt. Das ist noch kein Edge, aber es ist die Voraussetzung, um überhaupt einen zu messen.

6. Der Code ist der billige Teil

Nichts in dieser Liste spricht gegen Sprachmodelle. Sie machen den Teil, der früher Tage kostete, in Minuten, und das ist ein Gewinn. Der teure Teil war nie der Code. Es war die Liste oben, und die entsteht aus Backtests, die schiefgegangen sind, aus Konten, die anders reagiert haben als der Tester, und aus Brokern, die anders zählen als der Prompt annimmt.

Wer selbst baut, hat mit der Liste das Wichtigste in der Hand. Wer sich die Runde sparen will, findet unter Zusammenarbeit die Validierung als eigenen Schritt vor der Entwicklung, mit einem Verdikt, das auch „kein Edge“ lauten kann. Das billigste Ergebnis ist ein Nein, bevor die Entwicklung beginnt, und für ein Nein braucht es weniger als eine Woche.

Vorbehalt

Dieser Text misst nichts Neues. Die Zahlen stammen aus den verlinkten Studien, die dort ihre eigenen Vorbehalte tragen, und aus zwei Kundenfällen, die eine Anekdote sind und keine Stichprobe. Die Messung zum Indikator-Konsens ist intern und noch nicht veröffentlicht. Sprachmodelle ändern sich schneller als dieser Text; welche der Fehler ein Modell in einem Jahr noch macht, wissen wir nicht. Was bleibt, sind die Fehlerklassen selbst, weil sie Eigenschaften von Backtests und Brokern sind, nicht von Modellen.

Rein statistische Auswertung historischer Marktdaten. Keine Trading-Empfehlung, keine Finanzberatung. Historische Performance ist keine Garantie für die Zukunft. Trading mit Hebelprodukten kann zum Totalverlust führen.