

# Seminararbeit

## Algorithmisches Handeln mit Cryptowährungen

Carl Heinrich Bellgardt

31. August 2021

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Vorwort . . . . .	3
<b>2</b>	<b>Planung/Theorie</b>	<b>4</b>
2.1	Methodik . . . . .	4
2.2	Arbeitsmittel . . . . .	5
<b>3</b>	<b>Praktische Umsetzung</b>	<b>6</b>
3.1	Datenbeschaffung . . . . .	6
3.1.1	Testdaten . . . . .	6
3.1.2	Echtzeitdaten . . . . .	7
3.1.3	Datenspeicherung (SQL und Feather) . . . . .	7
3.2	Datenverarbeitung . . . . .	8
3.2.1	Trend . . . . .	8
3.2.2	Indikatoren . . . . .	8
3.2.3	Beispiel am MACD . . . . .	8
3.3	Auswertungs-/Bewertungsalgorithmus . . . . .	9
3.3.1	Ergebnisse der Indikatoren verarbeiten . . . . .	9
3.3.2	Ausgabe (GUI) . . . . .	9
3.4	Testen des praktischen Ergebnisses . . . . .	9
<b>4</b>	<b>Auswertung der Arbeit</b>	<b>9</b>

# 1 Einleitung

## 1.1 Vorwort

„Ein Risiko entsteht, wenn du nicht weißt, was du tust.“<sup>1</sup> Dieses Zitat stammt von Warren Buffet, einem der größten Investoren der Welt, und es bezieht sich auf den Aktienhandel, ein Themengebiet, das für seine Komplexität und Vielschichtigkeit bekannt ist, heutzutage wohl mehr, als jemals zuvor. Das Zitat gilt auch heute noch, aber für mich stellt sich die Frage: Wann weiß ich, was ich tue?

Da ich seit einiger Zeit ein großes Interesse an Finanzen und insbesondere dem Aktien- und Währungshandel habe, ergibt sich für mich die Umsetzbarkeit des obigen Zitates, wenn ich Vorgänge der Entscheidungsfindung im Aktien- und Währungshandel automatisiere und optimiere.

Im Bereich des Aktienhandels (Trading) gibt es bereits eine Vielzahl mathematischer Vorgehensweisen zur Berechnung von Faktoren und Strategien. Oft kommen diese Ergebnisse allerdings nur bei der Visualisierung zum Einsatz und werden eher als eine Art Handelsempfehlung gesehen. Das Vertrauen in eine Maschine ist geringer als in einen Menschen, zu groß ist die Angst vor einem übersehenem Faktor, auf den der Computer nicht programmiert ist.

Maschinen existieren nur zu einem Zweck: Dem Menschen die Arbeit zu erleichtern. Das gilt auch für Computer. Sie sind Rechenmaschinen der besonderen Art, denn Sie rechnen in einem Tempo, bei dem kein Mensch mithalten kann. So haben Computer in allen Bereichen unseres Lebens Fuß gefasst, im Alltag ebenso wie in der Industrie, bis hin zu Weltraumreisen. Der Nachteil aller Maschinen ist jedoch bis heute, dass sie nicht denken können, dass jede Maschine auf nur eine oder wenige verschiedene spezifische Aufgaben eingerichtet ist.

---

<sup>1</sup>manager magazin. Kai Lange und Christoph Rottwilm. Abgerufen 6. April 2021, von <https://www.manager-magazin.de/finanzen/artikel/erfolg-reichtum-glueck-denken-wie-warren-buffett-a-1057702.html>

Ich selber habe schon früh begonnen, Aufgaben an selbst geschriebene Programme oder Macros zu übergeben, um mir die Arbeit zu erleichtern. Manchmal habe ich nur automatisiert Dateien verschoben und manchmal hat mein Script an der Bildschirmfarbe erkannt, welche Taste es im Computerspiel drücken muss. Auch kleine Simulationen, wie z.B. der Test von Roulettestrategien lassen sich in wenigen Sekunden problemlos mehrere Millionen Male durchführen. Computer sind schnell und effizient.

Ist es also denkbar, auf Bauchgefühl und Intuition, also auf Denken und Erfahrung zu verzichten und stattdessen nach einem festen Muster, also einem Algorithmus, zu handeln. Kann man sich auf Indikatoren und reine mathematische Formeln verlassen? Ziel der folgenden Arbeit ist es, herauszufinden, ob anhand von Indikatoren eine erfolgreiche Handelsstrategie umsetzbar ist.

## 2 Planung/Theorie

### 2.1 Methodik

Der Versuch ist dann geglückt, wenn die gefundene Strategie in der Lage ist, im Durchschnitt einen Gewinn zu erwirtschaften, ohne dabei ein hohes Risiko zu tragen. In meinem Vorgehen bediene ich mich vieler bekannter Indikatoren und Strategien, ich werde diese per Backtesting<sup>2</sup> und Echtzeit-Simulation testen und vergleichen. Anhand der Testergebnisse entwickle ich eine eigene Strategie in Form einer Kombination verschiedener Indikatoren, so, wie es beim manuellen Handeln üblich ist. Zuletzt werde ich die Strategie auch praktisch anwenden, und zwar auf dem Cryptomarkt im Bereich der Cryptowährungen<sup>3</sup>. Dafür habe ich mich entschieden, weil einerseits dazu große Datensätze, die zum Testen der Strategie notwendig sind, vorhanden und mir zugänglich sind, und andererseits wegen geringer Nebenkosten gut

---

<sup>2</sup>Simulation mit historischen Daten.

<sup>3</sup>Währungen, die nur digital als Zahlungsmittel existieren. Meist unabhängig von Kontrolle (siehe Blockchain, Bitcoin)

mit Kleinstbeträgen zu handeln ist. Außerdem ist der Cryptomarkt ein ausschließlich digitales Phänomen, weshalb es bereits einige Programmierbibliotheken und APIs<sup>4</sup> gibt.

Meine Strategie, also meine in der Versuchsreihe erzielten Ergebnisse, werden einer Reihe Performencetests unterzogen, um sie hinsichtlich verschiedener Faktoren, wie zum Beispiel dem durchschnittlichen Gewinn oder möglichen Risiken, zu bewerten und gegebenenfalls modifizieren zu können.

## 2.2 Arbeitsmittel

Bei der praktischen Umsetzung meiner Arbeit werde ich mich der Programmiersprache Python bedienen, da ich darin bereits einige Erfahrungen gesammelt habe und die Sprache eine Arbeit mit zu erwartenden großen Datenmengen gut bewältigen kann.

Zu aller erst gilt es, die Grundlage für die Programmierarbeit zu bereiten. Dabei teilt sich das Programm in verschiedene Komponenten. In der Informatik wird oft das EVA-Prinzip erwähnt. Es steht für Eingabe, Verarbeitung und Ausgabe. Diese Unterteilung lässt sich bei nahezu jedem Programm anwenden, wie auch bei diesem. Die Eingabe befasst sich in diesem Fall mit einer wichtigen Frage: Woher beziehe ich Cryptodaten? Echtzeitwerte kann man relativ einfach mittels Webcrawling oder diversen Trading APIs beziehungsweise Python Libraries erhalten. Für umfangreiches Testen sind allerdings möglichst genaue Langzeitdaten nötig. Es bietet sich Kaggle an, ein Datenbankarchiv in dem sich unter anderem auch minütliche Daten hunderter Cryptos seit ihrer Aufzeichnung befinden, bei einigen Währung mehrere Millionen Angaben. Nun wo die Eingabe, also der Input, gesichert ist, geht es an die Verarbeitung. Diese beschäftigt sich mit allen Berechnungen, Indikatoren

---

<sup>4</sup>„Programmierschnittstelle“; ein Programmteil, meist öffentlich verfügbar, zum Interargieren mit anderen System, in diesem Fall Beispielsweise Datenbanken oder digitale Handelsplattformen

und Prognosen, etc., also dem eigentlichen Inhalt dieser Arbeit, dem Hauptteil. Dieser wird voraussichtlich komplett in Python programmiert, weitere Planungen sind dafür allerdings nicht nötig. Ein eher Zeitaufwendiger Teil ist zuletzt die Ausgabe. Einfach Zahlen auszugeben ist zwar einfach aber nicht immer gut für einen Menschen zu begreifen, weshalb eine gute Ausgabe nicht unwichtig ist. Die Python-Bibliothek Matplotlib eignet sich gut, um Graphen zu erstellen, da Graphische Oberflächen in Python allerdings viele Probleme mit sich bringen, werde ich anstatt einer GUI<sup>5</sup> einen Http-Server nutzen. Flask ist eine leicht zu bedienende API, welche alles mit sich bringt, was für das Hosten von Webseiten nötig ist. So wird die Seite letztlich in HTML, CSS und JavaScript geschrieben, was in den meisten Fällen sehr viel simpler ausfällt als eine tatsächliche GUI.

## 3 Praktische Umsetzung

### 3.1 Datenbeschaffung

#### 3.1.1 Testdaten

Die Testdaten sind jene Daten, die für die umfangreichen vorläufigen Tests verwendet werden. Nach dem alten Prinzip die Vergangenheit zu nutzen, um in der Zukunft bessere Entscheidungen zu treffen, werde ich meinen Algorithmus mit Daten aus der Vergangenheit erproben, bevor ich ihn auf Echtzeitdaten anwende.

Bei den Testdaten ist besonders wichtig, dass sie nicht nur möglichst genau und in gleichen möglichst kleinen Zeitabständen vorhanden sind, sondern auch, dass es möglichst viele sind. Im allgemeinen gilt, je mehr Daten vorhanden sind, desto aussagekräftiger sind auch die Ergebnisse.

Wie bereits unter den Arbeitsmitteln aufgeführt beziehe ich große und präzise

---

<sup>5</sup>Graphical User Interface (engl. Grafische Benutzeroberfläche)

Cryptodaten der Vergangenheit von der Datensatzbibliothek Kaggle<sup>6</sup>. Die Daten können direkt als Zip-Archiv heruntergeladen werden. Darin befinden sich Zahlreiche Dateien, alle nach dem gleichen Prinzip benannt. Ungetrennt von einander stehen im Dateinamen Cryptowährung und Referenzwährung, meist US-Dollar (usd), im Microsoft-Excel Format. Dieses ist verhältnismäßig langsam, kann aber mit der csv-Bibliothek für Python problemlos eingelesen und umstrukturiert werden.

### **3.1.2 Echtzeitdaten**

### **3.1.3 Datenspeicherung (SQL und Feather)**

Um große Datenmengen zu verwalten, gilt es, sich auf zwei Wesentliche Faktoren zu konzentrieren: Wie effizient und schnell können Daten gespeichert und geladen werden, und wie effizient können sie nach Kriterien gefiltert oder sortiert werden. Für eher kurzzeitige Speicherungen habe ich mich für die Python-Bibliothek Feather entschieden, da sie eine sehr gute Geschwindigkeit beim speichern und lesen von Daten bietet, aber auch weil sie das Pandas Format verwendet, mit dem man ohne weitere Umstrukturierungen gut weiterarbeiten kann, und welches auch direkt mit Zeit verknüpfte Daten unterstützt. Um gut sortieren und filtern zu können bediene ich mich dem weit verbreiteten Datenbanksystem SQL, in diesem Fall SQLite3. Im Gegensatz zu den meist benutzten Server und Client basierten Varianten wie MySQL benutzt SQLite nur eine lokale Datei als eigentliche Datenbank. Um Multithreading Fehler vorzubeugen, verwende ich es aber als einen lokalen Socketserver. Wie genau dieser funktioniert werde ich nicht erläutern, da es nicht für die Arbeit relevant ist. Es reicht zu wissen, dass der Server SQL-Anfragen zum Beispiel mit Suchkriterien annimmt und entsprechend Daten zurückgibt oder neue Daten einträgt beziehungsweise alte löscht.

---

<sup>6</sup>Kaggle. <https://www.kaggle.com/>

## 3.2 Datenverarbeitung

### 3.2.1 Trend

### 3.2.2 Indikatoren

### 3.2.3 Beispiel am MACD

Am Beispiel des MACD Indikators erkläre ich, wie ich einzelne Komponenten für die Datenverarbeitung erstelle. Diese Komponenten sind kleine Teilprogramme, die einen Indikator berechnen und Werte zurückgeben, die dann zusammen über die Tradingstrategie entscheiden. Der MACD (zu Deutsch: Indikator für das Zusammen-/Auseinanderlaufen des gleitenden Durchschnitts<sup>7</sup>) ist recht simpel zu berechnen. Er beruht auf dem Häufig genutzten EMA (exponential moving average), einem gewichteten Durchschnitt. Eine solche Rundungsfunktion hilft kleine Schwankungen herauszufiltern und den allgemeinen Trend sichtbar zu machen. Der MACD besteht aus einer MACD-Linie, welche sich aus der Differenz eines schnellen EMAs, meist aus den Werten von 12 Tagen, und eines langsamen, meist 26 Tage. Dann gibt es die Signal-Linie, welche ein EMA von meist 9 Tagen der MACD-Linie ist, also eine erneut geglättete Linie.

Bei der Auswertung des MACDs liegt die Aufmerksamkeit in der Differenz der MACD-Linie und der durch die erneute Glättung langsamer verlaufenden Signal-Linie, beziehungsweise ihren Schnittpunkten. Um diese hervorzuheben wird gerne die Differenz beider Linien errechnet, diese wird als Histogramm bezeichnet. Ist das Histogramm positiv, so indiziert sie einen Aufwärtstrend, ist es negativ, so handelt es sich vermutlich um einen Abwärtstrend. Wechselt das Histogramm sein Vorzeichen, also schneiden sich MACD und Signal-Linie, so ist dies ein neues Kaufsignal. Je höher der Betrag des Histogramms, desto stärker ist der Trend.

Bei dem MACD Indikator können also zwei Sachen abgelesen werden. Zum

---

<sup>7</sup>Wikipedia. Abgerufen 20. August 2021, von <https://de.wikipedia.org/wiki/MACD>



einen können durch Schnittpunkte Kaufsignale errechnet werden, zum anderen können die Werte des Histogramms und sein Verlauf als Trendbeschreibend verstanden werden. Ich werde beide Anwendungsmöglichkeiten des MACDs testen.

### **3.3 Auswertungs-/Bewertungsalgorithmus**

#### **3.3.1 Ergebnisse der Indikatoren verarbeiten**

#### **3.3.2 Ausgabe (GUI)**

### **3.4 Testen des praktischen Ergebnisses**

## **4 Auswertung der Arbeit**