MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„AIS – a Metric for assessing the Impact of an Influencer's Twitter Activity on the Price of a Cryptocurrency"

verfasst von / submitted by

Kevin Miller, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2022 / Vienna 2022

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 926
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Wirtschaftsinformatik UG2002
Betreut von / Supervisor:	Univ.-Prof. Dr. Uwe Zdun
Mitbetreut von / Co-Supervisor:	Ing. Dipl.-Ing. Dr. techn. Kristof Böhmer, BSc MSc

Contents

List of Figures

List of Tables

1 Introduction & Motivation

Cryptocurrencies have ushered in a new era of financial assets and financial freedom. Being able to partake in a novel financial system, one without intermediaries, in a trustless and anonymous fashion led to a widespread adoption of blockchain technology within only a little over a decade. Those who got in early were rewarded with ROI’s thought to be impossible in traditional financial markets. The stories of early adopters becoming millionaires or even billionaires over the span of a couple of years has led to controversy and jealousy, while the high volatility of the asset class discredits arguably groundbreaking and revolutionary technology and places it in a similar light of gambling.

The 2017/2018 bull-run was characterized by ICO’s (Initial Coin Offerings), followed by a collapse eerily similar to the dot-com crash in the 2000’s. Founders with nothing more than a whitepaper and a promise were able to raise millions of dollars to develop projects which turned out to be nothing more than a well selling idea. A lot of people made a lot of money, and even more people lost a lot of money. The hype and general interest about cryptocurrencies cooled down as the bubble burst and asset prices came crashing down in spectacular fashion. A lot of people left the space, many believed that cryptocurrencies would never recover. From its peak in the first week of January 2018, where the entire cryptocurrency market reached a market capitalization of roughly $ 760 Billion, the market lost about 88% of its value over the next year, bottoming out at $ 92 Billion [1]. This decline can be seen in Figure 1.

Figure 1: Crypto Total Market Capitalization calculated and displayed by TradingView[1] , shows the timespan of August 2017 until February 2019, 1 candle representing 1 week

Roughly two years after the previous crypto market peak, news about a pandemic would start spreading, sending the world into a state of panic. Covid-19 would cause global lockdowns and mandatory social distancing wreaked havoc on many industries, threatening a global recession of unforeseen proportions. To prevent total economic collapse, central banks and the US federal reserve started to print

[1] https://www.tradingview.com/ [accessed May 2022]

money to artificially prop up a heavily strained economy, and it worked. With a lot of new money entering the financial system and nowhere to spend it on, risk-heavy asset markets and other investment vehicles saw astronomical gains. The S&P 500 and the Nasdaq saw gains of upwards of 100% since they had crashed after the pandemic had started. [1] The real estate market has seen similar growth, with apartment prices in Vienna for example rising an astonishing 40% on average just over the past 2 years. [2] The largest gains were however once again experienced in the crypto market, where the whole asset class experienced unbelievable gains of over 2.600% from the lows of March 2020. The reason this is important, is because a lot of gains attract a lot of attention. If something quickly rises a lot in value, people - including very influential individuals - get interested and mainstream media starts picking up on it. In Figure 2 you can see the development of the global Google search trends for cryptocurrency from the beginning of 2020 until May 2022.

Figure 2: Global Google search trends for "cryptocurrency" from Jan 2020 until May 2022

Aside from meaningful technological advances in the space driving adoption forward, the main driver behind this wave of interest was the social media activity by Tesla-CEO Elon Musk. Musk started publicly stating his liking of Dogecoin, a cryptocurrency technologically very similar to Bitcoin, but created as a Meme by Jackson Palmer[2] and Billy Markus[3] in 2013. The founders state that Dogecoin was always intended as a joke, as a fun cryptocurrency with very low transaction fees and a massive number of coins (over 132 billion) to create an artificially low price tag. They never intended Dogecoin to be a used as a serious transaction vehicle. Palmer distanced himself from the project back in 2015 and is known to be somewhat of a cryptocurrency cynic [3]. Markus is also no longer directly affiliated with Dogecoin but remains a public figure within the cryptocurrency space, having roughly 1.5 million Twitter followers. The distancing of the founders from the project did not deter Musk from tweeting about Dogecoin being the best cryptocurrency for transactions and calling himself the "Dogefather".

[2] https://twitter.com/ummjackson [accessed May 2022]
[3] https://twitter.com/BillyM2k [accessed May 2022]

Figure 3: Global Google search trends for "Dogecoin" from Jan 2020 until May 2022

Musk furthermore mentioned that he had been working with the remaining developers of Dogecoin on improving various aspects like transaction speed and fees. All these news about a meme-loving multibillionaire endorsing such a project led to understandable sudden traction of the project. Between January and May of 2021, Musk was very vocal about his appreciation of the currency. This led to a spike in search trends (Figure 3) and also a spike in the price of Dogecoin.

Figure 4: Dogecoin price, calculated by Tradingview. Measured from Week of January 25th 2021) until Week of May 3rd 2021

The asset had risen in price by of 15.000% in the matter of months. After appearing in the American TV-show Saturday Night Live on May 7th, Musk stopped his vocal actions, causing the price to fall alongside. It had become apparent that the rise and fall of Dogecoin was clearly directly attributable to the activities of Elon Musk on Twitter. All of this price action is shown in Figure 4, where the price top coincides with Elon Musk's appearance on TV. The portfolios of investors are directly impacted by influencers that, knowingly or unknowingly, can change prices at their fingertips. In this paper we aim at identifying and quantifying individuals, which appear to have a strong link between their Twitter behavior and the price of an asset, giving them an Asset Influence Score (AIS), which aims at quantifying the suggested influence a Twitter user has over the price of an asset.

2 Problem Description

The goal of the establishment of Securities and Exchange Commission (SEC) by the US Congress in 1934 was to eliminate stock market manipulation. While this certainly proved effective to some degree, lesser regulated areas like over-the-counter (OTC) trading and emerging financial markets (like cryptocurrencies) are still heavily impacted by manipulation.

Manipulation can occur in many ways, from the actual purchase of assets to drive up prices to the release of information or the spreading of rumors. While the former typically requires a lot of capital, the latter can be executed by influential individuals without risking any capital, just by making public statements [4]. There is a fine line to be drawn between devious manipulation and the release of genuine information.[5] If the founder of a project tweets about a new partnership, an update to the project's roadmap, an upcoming update or anything else that the market participants deem relevant for the price of the asset, the degree of truthfulness is oftentimes second nature. Markets tend to react instantly to new information or opinions provided by influential individuals. It is therefore important for investors and market participants to recognize and monitor the activities of these individuals, also fittingly titled by Michael Cary as Crypto-Tastemakers [6].

One such tastemaker might be the head of a venture capitalist firm, openly invested in various crypto projects. Another one might be a well-respected businessman, the CEO of a fortune 500 company, the founder of the crypto project in question, a politician, a social media personality, a YouTuber[4], a famous sports player or even just a pseudonymous internet personality with a cult following. The background of the tastemaker can be deemed as rather irrelevant, what matters is how the market values their opinion.

The degree of truthfulness of information publicized via microblogging platforms like Twitter can oftentimes only be determined after the effect has already taken its course. In this thesis we therefore focus on the issuer of the message, rather than the exact message contents that might cause the price action. An argument can be made that the messenger plays a much more vital role when it comes to spreading an opinion compared to the actual message [7] [8]. We therefore plan on quantifying and analyzing the suggested influence that individuals on Twitter have over the market. Many very influential people can state their highly regarded opinion freely on social media, potentially causing unforeseen consequences for market participants. To our knowledge, not a single metric exists which quantifies users based on their suggested influence over asset prices.

[4] https://www.youtube.com/ [accessed May 2022]

3 Goals & Research Questions

Having established that many assets display price action that heavily correlates to public statements made by tastemakers, our goal is to create a transparent and replicable metric that quantifies this interrelationship. We therefore propose the **Asset Influence Score** or **AIS** for rating Twitter users based on the frequency with which their tweets occur in periods of abnormal price action. What exactly qualifies a price action period to be identified as abnormal will be discussed in detail in later chapters, but in general, a period can be identified as abnormal as soon as its trading activity exceeds the average trading activity by a certain factor.

The AIS represents the certainty of a user's newly issued tweet being issued within a period of abnormal price action.

We deliberately choose to stray from stating that tweets *cause* certain trading activity, but rather point out strong correlations which could indicate causation. It goes without saying that the price of an asset is not solely dependent on the public opinion of some twitter user(s). Factors like general market sentiment, trading volume and macro-economic factors like political instabilities, wars, sanctions, or inflation all play a role in forming the price of an asset at any given time. It is nearly impossible to incorporate and analyze all variables, but one can approximate a good-enough solution given the capabilities.

We will accomplish this task by combining price data and twitter activity to identify users that warrant the assumption of having caused abnormal trading activity through their tweeting behavior. Identifying these individuals is useful in many ways. It can help market participants make more informed investment decisions and potentially even avoid of fraudulent behavior, as has been displayed by countless pump & dump schemes [9] over the past couple of years, especially in the cryptocurrency space. A good example would be the pumping & dumping of Ethereum Max, which was promoted by A-List celebrities like Kim Kardashian and Floyd Mayweather and resulted in a class action lawsuit [10]. It would've become apparent that two individuals with absolutely no history in cryptocurrency are clearly being paid to talk about the currency to pump the price of the asset, as their posts coincided with direct price movement.

On the contrary, it also gives investors an idea of who to watch and which persons opinions are weighted heavily when it comes to genuine investment opportunities. Knowing who and to what degree someone has substantial influence over the price of an asset can be invaluable to many people, as their personal finances are directly exposed to another person's opinion.

3.1 Research questions

While conducting the research for the theoretical basis of this thesis will we answer the following questions. Research Questions (RQ) 1 - 4 were formulated to be answered in a way that gives the best possible overview over the current state of the art in the field of price prediction via Twitter. Question 5 will be answered at the end of this thesis by conducting our own research on the topic.

RQ 1. What are current approaches and algorithms for price prediction that utilize data from microblogs and are there any areas that lack research?
RQ 2. How and to what extent do price predictions utilize sentiment analysis?
RQ 3. How is the data fetched, prepared and processed and how can it be presented?
RQ 4. Are the results meaningful, explainable and replicable?
RQ 5. Is it possible to create a meaningful metric for the classifying the potential influence of a given user over a given asset?

To answer questions 1 – 4, which revolve around the current state of the art in the field of microblog analysis and price prediction, we will utilize a well-established form of literature analysis called a systematic literature review. Question number 5 will be answered applying principles of design science described by Hevner et. al. [11]

4 Systematic Literature Review

This systematic literature review is conducted according to a defined process based on Barbara Kitchenham [12]. Kitchenham documents the following phases and activities that form the systematic literature review process:

- Phase 1: Planning the review
  - Identification of the need for a review
  - Development of a review protocol

- Phase 2: Conducting the review
  - Identification of research
  - Selection of primary studies
  - Study quality assessment
  - Data extraction & monitoring
  - Data synthesis

- Phase 3: Reporting the review
  - Reporting the results of the review

4.1 Identifying the Need for a Review

The first step was to evaluate whether any systematic reviews had already been performed which cover this exact or any similar areas of research. Unsurprisingly, the field of asset market prediction using various forms of social media data is well explored, as there is a lot of money to be made by discovering effective means of price prediction. The most predominant way of attempting to predict asset movements is by using various forms of sentiment analysis, followed by event detection, which usually only builds upon sentiment analysis methods. These two methodologies make up the lion's share of published research regarding asset price prediction and have also proved themselves as being quite effective at doing so. [13]

We also focus this review solely around the microblogging platform Twitter. The reason for this is a combination of factors. Twitter has established itself as the primary platform for influential people to express their opinion on various topics. Microblogging platforms encourage the user to post quick updates without giving them too much thought. It is intended to be quick and easy and allows for a more unfiltered and natural insight into somebodies thoughts [14]. With 330 million monthly active users, it is also the most popular microblogging worldwide [15]. Public figures value their presence on Twitter, as it gives them an easy medium to quickly and effectively communicate with a large number of people. This makes it the perfect platform to analyze the activity and influence of "real" individuals. Unlike news outlets or blogs, a Twitter account is typically just run by the individual in question, making it a well-suited choice for this study.

4.2 Search Strategy

This section of the paper describes the process and parameters of the literature research. It intends to give a concise overview of what databases and portals we used and how they were searched, as well as the decision process involved in curating the search results.

4.2.1 Resource Selection

We chose two separate computer science research databases to conduct our research, those databases being Web of Science[5] (WoS) and Scopus[6]. [16] mentions these databases as good performers in terms of indexing count and quality. These databases are also accessible through the University of Vienna. Additionally, both databases feature an elaborate querying mechanism, which is essential for close replicability of search results.

Other databases considered were DBLP[7] and Google Scholar[8]. Even though DBLP was also mentioned as being a strong performer in terms of indexing, its comparably extremely poor querying capabilities (only basic boolean operators) made us rule out DBLP as a research database. Google Scholar only allows for queries that match the paper's title or anywhere in the text, not explicitly its abstract or keywords, which also ruled it out.

4.3 Study Selection

The following segment discusses the selected criteria that determined the inclusion or exclusion of given research papers regarding this literature review. We will explain how we filtered through the search results to come up with a curated list that will function as the basis for the review.

4.3.1 Inclusion criteria

These criteria must be met for the literature to be eligible for further consideration. The absence of any of these criteria will result in the paper being discarded from the set of research considered for this literature review.

- The title, abstract or the keywords of the paper in question must match the conditions of the query string
- The paper is indexed and available through either Scopus or WoS
- The paper in question must describe primary research
- The paper was either published open access or is accessible through the University of Vienna
- The paper must focus on the analysis, modeling or prediction of tweets in regard to asset prices using Twitter data

[5] https://www.webofknowledge.com [accessed April 2022]
[6] https://www.scopus.com [accessed April 2022]
[7] https://dblp.org/ [accessed April 2022]
[8] https://scholar.google.com/ [accessed April 2022]

4.3.2 Exclusion criteria

Should any of the following criteria be met, the paper is also excluded from the literature basis.

- The paper is not available in its entirety
- The paper is not accessible through the University of Vienna
- The paper is not written in English or German
- A more recent version of the paper that meets the inclusion criteria and fades the exclusion criteria is available
- The paper violates any of the inclusion criteria

4.4 Keywords and Query String

A query string was created to cover all topics relevant to the researched field. It was then adapted syntactically to the respective database but was kept semantically identical for all database queries. We started with the keywords "twitter" and "cryptocurrency", henceforth called Query String 1 (QS1) which returned a total of 69 results for Scopus, and a total of 46 results for WoS.

(twitter ∧ cryptocurrency)

As these results were still subject to further filtering and selection, we determined that the number of results was insufficient for conducting a literature review. After some thought we figured that some valuable insight is likely to be gained by incorporating research that focuses on the stock market. As cryptocurrencies and stocks share strong similarities in terms of accessibility, the way they are traded and the exposure to human emotion, it only made sense to include research about the stock market into the literature basis as well. We therefore extended QS1 with the keywords "stock", "asset", "price" and "market capitalization", resulting in the following QS2:

twitter ∧ (cryptocurrency ∨ stock ∨ asset ∨ price ∨ market cap)*

QS 2 returned a total of 951 results for Scopus and 615 results for WoS. We then finally wanted to make sure that all papers that only cover specific cryptocurrencies were also included, therefore we extended QS2 with the Top 10 non-stablecoin (not pegged to the value of an underlying fiat-currency) cryptocurrencies, ranked by their market capitalization. At the time of conducting the review (early May 2022), these cryptocurrencies were: Bitcoin, Ethereum, BNB, XRP, Solana, Cardano, Terra, Dogecoin, Polkadot and Avalanche. These additions resulted in the following, final Query String (QS3):

twitter ∧ (cryptocurrency ∨ stock ∨ asset ∨ price ∨ market cap ∨ Bitcoin ∨ Ethereum ∨ BNB ∨ XRP ∨ Solana ∨ Cardano ∨ Terra ∨ Dogecoin ∨ Polkadot ∨ Avalanche)*

QS 3 yielded a total of 1,009 results for Scopus and 651 results for WoS. We then also decided to limit the results to research published open access in the English or

German language as a conference paper or an article. When applying these constraints, Scopus returned 265 papers for QS 3, while WoS returned a total of 217 papers. Exact statistics can be seen in Table 1.

	Scopus	WoS
QS 1	69	46
QS 1 w/c	22	22
QS 2	951	615
QS 2 w/c	249	201
QS 3	1,009	651
QS 3 w/c	265	217

Table 1: The number of papers found with the queries in the respective database. w/c = with constraint

4.5 Literature Search

The exact query strings (QS) from Table 2 were run on the chosen databases, with (w/c) and without the aforementioned constraints (English language, conference paper or article). The query string tried to find matches in either the title, the keywords or the abstract of the paper. The constraint string was added with a logical AND operator.

	Scopus	WoS
QS 1	TITLE-ABS-KEY ("twitter" OR "social media" AND "cryptocurrency")	TS= (("twitter" OR "social media") AND ("cryptocurrency"))
QS 2	TITLE-ABS-KEY ("twitter" AND "cryptocurrency" OR "stock" OR "asset" OR "price" OR "market cap*")	TS= (("twitter") AND ("cryptocurrency" OR "stock" OR "asset" OR "price" OR "market cap*"))
QS 3	TITLE-ABS-KEY ("twitter" AND "cryptocurrency" OR "stock" OR "asset" OR "price" OR "market cap*" OR "Bitcoin" OR "Ethereum" OR "BNB" OR "XRP" OR "Solana" OR "Cardano" OR "Terra" OR "Dogecoin" OR "Polkadot" OR "Avalanche")	TS= (("twitter") AND ("cryptocurrency" OR "stock" OR "asset" OR "price" OR "market cap*" OR "Bitcoin" OR "Ethereum" OR "BNB" OR "XRP" OR "Solana" OR "Cardano" OR "Terra" OR "Dogecoin" OR "Polkadot" OR "Avalanche"))
constraint	Doctype (cp OR ar) AND Language(english)	LA= (English) AND DT= (Article OR Proceedings Paper)

Table 2: The exact queries ran on the respective database

Both Scopus and WoS have an export feature, so both result sets were then exported in the RIS format and imported into Zotero[9] as a combined dataset with a size of 482 entries. This made it easy to spot and remove duplicate entries, which was done manually within Zotero. For any duplicate entry the entry that contained a higher quantity of/ more accurately filled meta information fields was kept in the final

[9] https://www.zotero.org/ [accessed June 2022]

dataset. These fields are for example the exact publishing date, the ISSN, an access URL or the DOI (Digital Object Identifier). We called the resulting dataset the raw dataset. A total of 157 duplicates were removed, leaving the raw dataset with 325 entries.

In the next step we examined the title and abstract of every paper in the raw dataset and excluded every paper that was deemed irrelevant or hardly relevant to our research focus. Common reasons for exclusion were the focus on company brand image instead of financial markets, corporate or social analysis, the demographic analysis of users or the study of public health. We also excluded all papers that did not describe primary research (if noticeable from examining title & abstract), as stated in our exclusion criteria. This process resulted in the removal of 299 papers, which left 53 papers remaining in the dataset which we called curated dataset. These 53 papers were then examined in their entirety, starting with the introduction and conclusion. If the research was deemed relevant after the full review it was accepted into the final dataset, which will serve as the basis for this literature review. During the full-paper review, a further 23 papers were removed. While performing the review and analysis of the papers, an additional 8 papers were incorporated, that were not initially found through the database queries. These papers were found through backward-search instead, which leaves a total of 38 papers in the final dataset. Table 3 gives an overview over the selection process, whereas Figure 5 gives a more graphical representation of the curation process.

Action	Dataset	Count
	Combined results of QS 3 w/c	482
Remove Duplicates		- 157
	Raw dataset	352
Title & Abstract Analysis		- 299
	Curated Dataset	53
Full-Paper Review		- 23
Backward-Search		+ 8
	Final Dataset	38

Table 3: Overview over the curation steps for the literature research

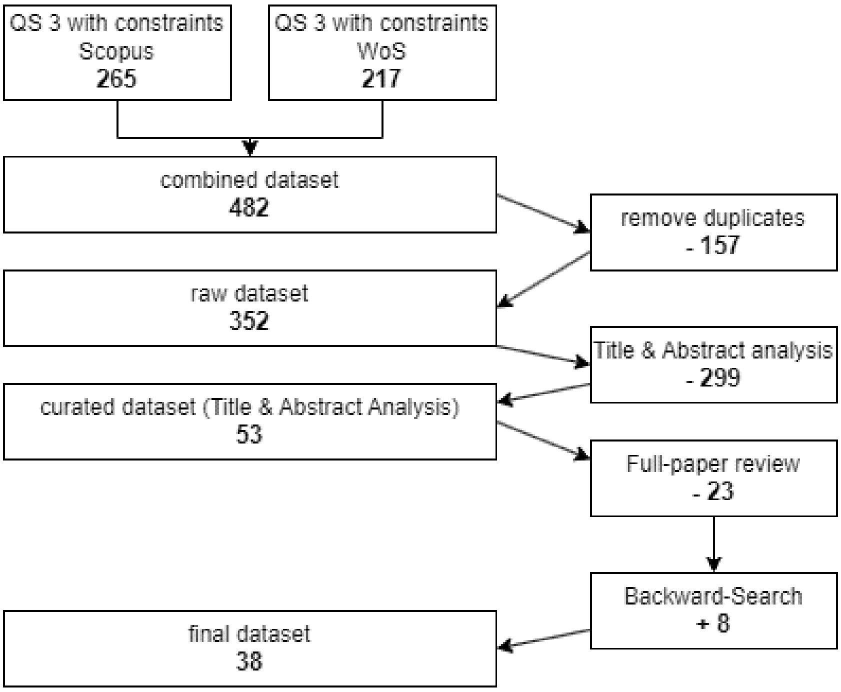

Figure 5: Graphical overview of the curation steps for the literature research

5 Results

5.1 Topic Discovery and Research Areas

The following section will elaborate on different topics that were identified within the literature basis. While all the research is somehow related to each other, we identified and divided the research into different topics to give a more structured overview over the available literature.

5.1.1 Sentiment Analysis & Price Prediction

It was long believed that asset markets follow the Efficient Market Hypothesis (EMH), which states that the price of a share or asset is the reflection of all available information, effectively stating that all information is priced in and that price movements follow the random walk theory, meaning that the results of infinite price predictions will always converge at 50% accuracy [17]. This is because news that regard an asset are unpredictable and consequently so is their impact on the price. While news most certainly play a role in influencing the price of assets, so does public mood and investor sentiment. [18] describes the economy as a complex system of human interactions rather than the pure result of economic fundamentals, stating that investors actions can be derived from their thoughts and feelings, which stem from the interaction with other human beings, therefore directly contradicting with the EMH [19]. Behavioral economists like Tversky and Kahneman further established that decisions that could be made on a completely rational basis are still impacted by the decision maker’s emotional state [20].

Much of the world's data exists in the form of unstructured text data in the form of emails, text messages, tweets or articles. This structureless data has raised the need to create a system capable of extracting valuable information, leading to the creation of natural language processing (NLP). NLP is collection of methods that allow computers to understand text and extract information. [21] In a nutshell, sentiment analysis works by first aggregating streams of data (in our case that stream of data are tweets). This data is then preprocessed (removal of punctuation, stop-words, URLs, lemmatization of content, etc.) and subjected to different toolkits, which analyze and gauge the overall sentiment of the content. The result is a classification of the input being deemed either negative, neutral, or positive. These toolkits contain large quantities of human-labeled data which have been given a certain sentiment score (for example "absolutely astonishing" could be weighted as +5, while "a little disappointing" could be weighted as -2). The result is then classified depending on the sum of all scores for either individual words, phrases or sentences. [22] If the result is positive, so is the sentiment, and vice versa.

Bollen et al. used sentiment analysis on large-scale Twitter feeds and mapped the determined sentiment to the Dow Jones Industrial Average (DJIA) by using a self-organized fuzzy neural network. They were able to make a price direction with 87.6% accuracy. They did however not focus on the influence of an individual user, but rather a broad and homogenous user group [22]. [23] extend this work by implementing a naive portfolio-management strategy based on their findings but fail to improve the prediction accuracy of Bollen et al.

Abraham et al. focused their research on tweet volume and Google trends, rather than sentiment alone. They gathered tweets from the Twitter API by querying for "#" followed by "Bitcoin" or "Ethereum" and combined the data with tweet volume collected via Tweepy[10] as well as Google trends data for the respective time frame. The authors used the VADER [24] system for sentiment analysis and presented their findings in expressive and easy to understand charts. They did find significant correlation between both Google trends data as well as tweet volume and the price of bitcoin but determined sentiment analysis to be a non-reliable indicator. The authors argue that positive sentiment is based on fundamentals like privacy and decentralization rather than price action, factors which don't change. Furthermore, tweet sentiment was analyzed to be mostly positive or neutral regardless of price, leading to the conclusion that sentiment was not a potent indicator for detecting market downturns.

These findings coincide with Choi et al.'s studies, which determined that autoregressive prediction-models that included Google trends data outperformed models without the data by 5 to 20 percent. The authors used Google trends data to predict *present* activity in various economic sectors like automobile sales or traveling and left actual future prediction up to future research.

Beck et al. map Twitter activity to recently published cryptocurrency-related news articles and explore various machine learning models to predict the popularity of the news article on Twitter. They use models such as linear extrapolation or linear and random forest autoregressive models. The authors provide a graphical

[10] https://www.tweepy.org/

representation of their prediction accuracy and had even deployed the trained AR model to Amazon Web Services in an interactive webpage, which is unfortunately no longer reachable [25]. Guo et al. utilize Twitter Sentiment Score (TSS) to predict market movements with 67.22% accuracy [26].

Bhuvaneshwari et al. proposed the Tolerant Flexible Coordinated Multi-Agent Deep Reinforcement Learning (TFCMA-DRL) model, which combines tweet sentiment and news data utilizing Extreme Machine Learning and Restricted Boltzmann Machines respectively. It then learns the price patterns through these indicators and forecasts future prices with promising results of 96.67% accuracy [27].

Mohapatra et al. have created KryptoOracle which analyzes real-time crowd sentiment with VADER and makes price prediction based the sentiment of the Twitter feed. The system is also capable of self-learning from incorrect predictions. The selling point is the Apache Spark based architecture which allows KryptoOracle to process a large volume of incoming data [28].

5.1.2 Classification

Alkubaisi et al. created a conceptual framework for sentiment analysis based on hybrid naïve bayes classifier. They sourced data from Twitter via the Twitter API and relied on expert labelling for the dataset. The authors do provide the pseudocode for their algorithm, but do not evaluate the efficiency of their approach nor do they compare it to other approaches. They also do not provide any data for replication, which is problematic as the “expert labeling process” is subjective, making the classification effectively irreplicable [29].

Ben-Ami et al. use unsupervised multi-view learning to mine a large corpus of messages used to train a domain-specific sentiment analysis model. They seed the learning process with a generic domain-independent SA system and identify correlation in stock price trends and terms used in Tweets to identify and extract relevant terms. The authors show that by using domain-specific SA they were able to increase prediction accuracy by up to 9,1% in the financial sector [30].

Vilas et al. study the effects of cashtag collision between publicly traded companies and cryptocurrencies. A cashtag on Twitter is the $-symbol followed by the ticker for the asset, usually three letters. An example of this would be $AMD for Advanced Micro Devices, Inc. or $BTC for Bitcoin. Cashtags are typically used as an efficient way to gather tweets related to a stock or asset. The authors identify and study the effects of homonym cashtags (cashtags that represent both a company and a cryptocurrency – e.g. $XLM for Stellar Lumens [Cryptocurrency] and XLMedia [company]). The authors identify a negative impact created by the emergence of cryptocurrencies by diluting financial information on Twitter through homonym cashtags. They tackle the problem by applying word-based heuristic filters to the Tweets on a document-level, allowing for a correct classification between company and cryptocurrency in 97% of cases [31][32].

5.1.3 Influencers and Single Users

Brans et al. analyzed the correlation of Trump's tweeting behavior and the stock price of companies mentioned in tweets. They sourced the data from thetrumparchive.com[11] and found no overall significant response from the stock market to the tweets of Trump. When accounting for sentiment, the authors found that the market reacted stronger to negative rather than positive sentiment towards a company displayed by the president [33].

Gjerstad et al. also found that markets on average reacted negatively and with increased volatility to Trump's tweets. The authors used Latent Dirichlet Allocation (LDA) to detect tweets affiliated with the topics "trade war" or "border security". They prove the efficiency of their system by implementing a simple trading strategy of holding the S&P 500 index, selling it whenever Trump tweets about said topics, and buying it back 75 minutes later, achieving above market returns [34].

5.1.4 Event Detection

Event detection usually involves monitoring and detecting sudden spikes in social media activity. Alostad performs event detection by monitoring the hourly tweet volume for certain stocks. She uses a sliding-window approach for the past 20 hours and trigger a breakout response if the mean of the current window including 2x the standard deviation exceeds that of the previous window. The breakout response is the subsequent sentimental analysis of news articles linked in tweets issued in the breakout period [35]. Her approach is somewhat unique, given that she doesn't perform sentiment analysis on the actual tweets, but rather uses spiking tweet volume to detect the emergence of a relevant news article, which is subsequently analyzed to perform a directional prediction. She also shows that document-level sentiment analysis does not yield a statistically significant boost in directional prediction accuracy.

De Arriba-Perez et al. use a three-layered machine learning architecture to try and detect what they called financial opportunities – clusters of tweets with positive sentiments – and achieve an accuracy of 83%, meaning that the price change was positive 83% of the time. They do however also focus on a broad, homogenous userbase to make these detections [36].

Hazem et al. incorporate the aspect of magnitude into their research. Instead of purely focusing on the question of whether financial announcements affect share prices, they also tried answering how much the price would be affected. This study however lacks concrete predictions and also does not focus on individual users [37].

5.1.5 Price Manipulation, Spam & Bots

Nizzoli et al. discovered 15 bot networks on Twitter which distribute invite links to pump-and-dump as well as ponzi-scheme groups on Discord and Telegram. The authors found that these networks are responsible for 75.4% of invite links to such channels. They did so by cross-checking over 50 million messages across Twitter,

[11] https://www.thetrumparchive.com/

Discord and Telegram and analyzing diffusion of invite-links, as well as using LDA to discover topics and bags-of-words related to such schemes [38].

Victor et al. studied pump-and-dump schemes coordinated on Telegram and executed on Binance, the world's largest cryptocurrency exchange. They obtained ground truth confirmation for pump-and-dump activity by pulling and confirming data from the Telegram API and the identification was done manually. The authors note that it is difficult sto separate genuine engagement from pump-and-dumps without manual verification [9].

5.2 Discussion

The literature review now allows us to answer research questions 1 – 4.

RQ 1. What are current approaches and algorithms for price prediction that utilize data from microblogs and are there any areas that lack research?

Current approaches almost exclusively use sentiment analysis, tweet volume and search volume or a combination of these data points to make predictions on price changes. They analyze sentiment from a broad twitter userbase or from a single user, but don't incorporate a subset of users which could be identified as opinion leaders.
Algorithms and models that are used include but are not limited to various forms of (convolutional) neural networks, VADER, autoregressive prediction models and Latent Dirichlet Allocation, with the best results achieved by utilizing neural networks.

RQ 2. How and to what extent do price predictions utilize sentiment analysis?

Price predictions commonly incorporate sentiment analysis, however many authors note that SA had a negligible effect on their model's performance. Some even go as far as to exclude SA results from their model, as they would perform better without. The best results were achieved by incorporating multiple data streams, such as aforementioned tweet volume, search engine volume or price action data. The research that bases their prediction solely on sentiment analysis, especially domain-agnostic SA, performed worse on average compared to ensemble data.

RQ 3. How is the data fetched, prepared and processed and how can it be presented?

The data was always fetched via the Twitter API (either through hashtag or keyword search) and in cases of other data streams either the Google API (for search trends) or free price data APIs for price data. The authors typically follow the rather standard procedure of transforming tweets to lowercase, stemming or lemmatizing the content and removing punctuation, stop words, emojis, URLs. No detailed explanations could be found about the preprocessing steps involved with other data streams.

The Twitter data was typically not presented graphically or in table form, but one could argue that is not feasible nor necessary due to the sheer amount of tweets used, but also due to ethical concerns around anonymity of the analyzed users. Many authors added figures to their research, showcasing the results of sentiment analysis, mapping sentiment/volume/search trends to price data, showcasing prediction accuracy etc., which were helpful for understanding their research.

RQ 4. Are the results meaningful, explainable and replicable?

Many authors do mention that their approach could achieve statistically significant results and predictions, claiming of being capable of achieving above market returns. The problem lies within the explainability and replicability aspects. In terms of explainability, the approaches achieving the best results (typically neural networks) lack explanation on how exactly the decisions were made. While the authors typically conducted and reported about experiments in different settings, few if any were able of explaining how their models were able to achieve results.
We also could not find a single paper which provided sample data or source code with which we could exactly replicate the author's claims. There is an argument to be made about ethical concerns regarding the individual user. As many authors note, Twitter users might not be aware that their opinions can be and are used and analyzed in a scientific setting. One could therefore argue that it is not morally acceptable to store uncensored data from Twitter users. Regardless of ethical concerns, we could not find any instances where authors would provide downloadable pseudonymous/ anonymous datasets with which their claims could be verified. Regarding source code, we also could not find any instances of open source availability of any models or approaches, only pseudocode references.

Findings

It has become apparent that existing research regarding price prediction via Twitter focuses very heavily on sourcing information from a large userbase. Rather than identifying opinion leaders/tastemakers, most authors source their information from "the crowd". There exists research which does focus on single users, but the only available literature focused on the actions of former US-president Donald Trump. These approaches are however still relevant and usually generally applicable, as the methodologies could've been performed on any arbitrary twitter user. To the best of our knowledge, research which focuses on a dynamic subset of individuals regarding price analysis does not or hardly exists.

Also, most prediction models, while providing detailed insight about conducted experiments, failed to explain how their models made its predictions. Another interesting and admittedly surprising discovery was the overarching agreement of authors about the fact that sentiment analysis alone is not a competent indicator at predicting future price action. While some authors, also in heavily cited and respected work, found that SA can in fact be used to make good predictions, many authors also disagree. This review leads us to claim that there exists a notable gap in research concerning opinion leaders and their suggested influence on the price action of cryptocurrencies.

6 Methodology

We will now analyze our proposed approach for quantifying the influence of tastemakers over the price of a cryptocurrency. We will start off by giving a basic overview of the process and follow up with detailed explanations about every step.

6.1 General Approach - Overview

This chapter will provide a high-level overview of the computation of the AIS. Variable definitions, functions and formulae have been omitted unless absolutely necessary. Every step will be explained in greater detail in the subsequent chapters.

Figure 6 represents a model which is intended to give a very basic birds-eye view of the steps involved. First, the user is required to enter the name of the cryptocurrency, a timeframe, the *sliding window size* and the *breakout threshold factor* (explained later), as well as a few other parameters. This is done using the graphical client application which was created to perform all the necessary data fetching, preparation and computation (see chapter 6.7 – Client Application).

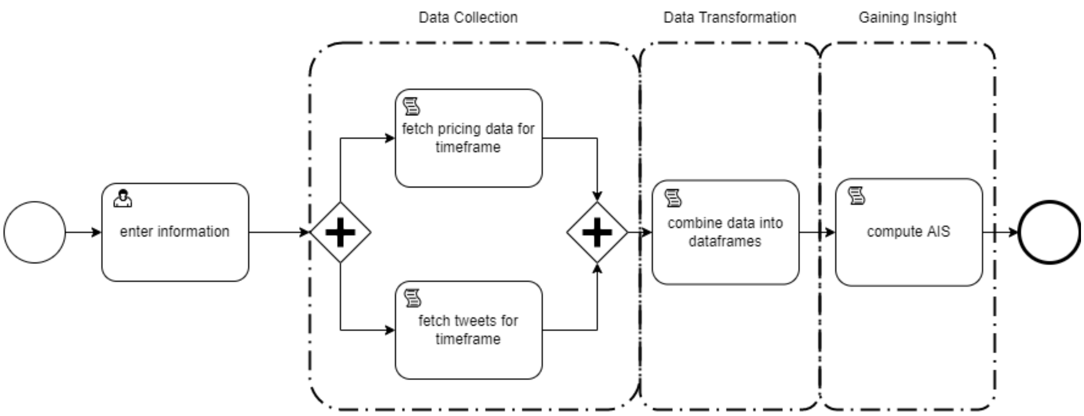

Figure 6: Proposed AIS calculation approach - basic overview

After the user has entered the required information, the application will fetch the data it requires automatically from external APIs. This data consists of hourly trading data (price and trading volume) and the top 100 tweets issued every hour, only for the timeframe specified by the user. These data points are collected separately from each other, as they require two different APIs to fetch. Furthermore, the data is stored in a raw format directly as the API returns it, meaning no transformation or extraction has been performed yet. We also cache this data locally to reduce the need for subsequent API calls.

We use hourly data, as it strikes a fine a balance between granularity and efficiency. Had we used larger time segments, it would have become increasingly more difficult to identify single users which might have caused a rise in trading activity, whereas had we used smaller time segments, the impact of a tweet might have not taken yet full effect before the end of the segment.

After the data is fetched, the application extracts the information which is required for the AIS from the previously fetched data and combines it into our custom data structure called a dataframe. A dataframe essentially represents an hour of market activity, containing the price action of the asset and the most relevant tweets issued during the hourly segment. We decided to introduce dataframes to allow for a very streamlined implementation of the subsequent calculation process. The application only leverages the information stored in dataframes to compute the AIS, so we are technically able to work with a multitude of data formats without impacting the calculation process. Upon initial creation, the dataframes are also cached locally to speed up subsequent executions.

To measure trading activity, we have introduced a variable called *velocity,* which captures the trading activity (price change and the corresponding trading volume) of one hourly time segment in a single number. Significant changes in price and/or trading volume will result in a higher velocity. The velocity of a single segment by itself is rather meaningless, but it gains a lot of meaning when compared to the velocity of other segments.

One of the pieces of information initially provided by the user is the *sliding window size*. The window size defines the length of a sliding window, over which the average velocity is computed. This approach was inspired by Alostad et. al. [39].

We have also introduced a variable called *magnitude,* which denotes by what factor a dataframe's velocity differs from the current sliding window's average velocity.

Dataframe	1	2	3	4	5	6	7	8
Price change	1$	1$	0.9$	1$	5$	1$	1$	7$
Volume	10	8	10	10	100	25	10	200
Velocity	10	8	9	10	500	25	10	1400
Average Velocity	-	-	-	9	9	172	177	178
Magnitude	-	-	-	1,11	55,50	0,14	0,05	7,86

Table 4: Demonstration of the Sliding Window-based Calculation of Average Velocity and Magnitude with a Window Size of 3

Table 4 shows an example of how the sliding window works. The window size is set to 3, so the first dataframe for which the average velocity and the magnitude can be calculated is dataframe 4 (or after 3 hours). We want to note that the first {window-size}-amount of dataframes are excluded from the AIS calculation to ensure full data integrity.

In our example, dataframe 5 shows a significantly increased velocity of 500 compared to the average 9, or a 55.5-times increase (magnitude). As mentioned previously, we analyze the changes in velocity compared to the average for an asset and timeframe. By calculating the magnitude, we can capture factor by which trading activity (velocity) changes, which is uniform and generally meaningful.
Another section we want to highlight in this example is dataframe 8. Here the magnitude is only 7.86, even though the velocity is a lot higher (1400) compared to dataframe 5 (500). This is due to the sliding window incorporating only the last 3 velocities. If e.g. some external event (like a tweet) kickstarted a long-lasting period

of increased trading activity, we still want to reliably identify abnormalities within that period. This is the reason why we use a sliding window rather than a global average.

Figure 7 shows an example of this. The upper line chart shows the price action, the lower chart shows trading volume. If you compare the activity in August to the activity in June, everything would be an abnormality. Using an appropriately sized sliding window, only the activity in the beginning of July will be identified as abnormal.

Figure 7: An example of a sudden general increase in trading activity shown in Tradingview

The user must specify the *Pearson Correlation Coefficient* (or *PCC*) of the cryptocurrency to Bitcoin (BTC), which can either be calculated or found on the internet[12]. The PCC describes, how heavily an asset is correlated to another asset. In the cryptocurrency space, pretty much all cryptocurrencies are significantly correlated to Bitcoin. This means that if Bitcoin's price moves, the prices of the cryptocurrencies correlated to BTC move in the same direction and with a similar magnitude. By incorporating the PCC, we can compare the magnitude of Bitcoin and the magnitude of the cryptocurrency being analyzed for each time segment. We can then subtract Bitcoin's magnitude (weighted by the PCC), to be left with the magnitude which was not caused by regular market moves. We call this *Magnitude attributable to External Factors* (*MEF*), whereby those external factors incorporate for example a tweet which could've caused that move.

In the next step, we iterate over every dataframe in the specified timeframe. For every user that issued a tweet (or multiple) within that dataframe's time segment, we keep score of the magnitude (MEF) and associate it with that user's tweets. When every dataframe has been analyzed, we can compute the average MEF associated with every user that has tweeted within the timeframe. This already gives

[12] https://cryptowat.ch/correlations [accessed July 2022]

us a rough idea of which users tweets frequently co-occur within time segments of abnormal magnitudes.

Furthermore, we introduced something we called *Engagement Share (ES),* which we use to attribute the magnitude of the time segment to each users' tweets. The reason we introduced this is so that we can differentiate between impactful tweets and copycats.

Rank	Text	Likes	Retweets	ES	MEF
1	SpaceX is going to put a literal Dogecoin on the literal moon	519100	49631	98,46	1.001,54

Figure 8: Example of a tweet with a very high Engagement Share

In Figure 8 you can see a tweet issued by Elon Musk about Dogecoin. It had an Engagement Share of 98.46%, meaning that 98.46% of all Likes and Retweets given to any tweet about Dogecoin in the hour that this tweet was issued in were given to this tweet. The MEF of the associated hourly segment was also 1,001, so Dogecoin-trading experienced a very heavy increase in velocity during this hour.

While this is still no solid proof that the tweet in this example had caused the sudden spike in magnitude, we believe it is very reasonable to assume so. We therefore attribute 98.46% (or roughly 984,6) of the MEF to Elon Musk, while the remaining 1.56% are attributed to the other tweets issued during this hour. This *attributable magnitude* is what allows us to separate non-tastemakers, which could have easily tweeted something Dogecoin-related during the same hourly segment, from tastemakers like Elon Musk.

One of the variables a user can specify is the factor (later referred to as *breakout threshold factor – BTF*), which a time segments magnitude must exceed to be considered an anomaly. This BTF is used as the criteria for the AIS.

The AIS describes the probability that on a given day, a specific user's newly issued tweet will co-occur in a time segment which's magnitude exceeds the BTF.

For example:

Given the timeframe from January 1, 2020, until December 31, 2021, Elon Musk's AIS with a BTF of 25 on December 31, 2021, for Dogecoin is 62,32 This means that a newly issued tweet from Elon Musk that is related to Dogecoin has a 62,32%-likelihood of co-occurring within an hour where Dogecoin-trading will experience a magnitude of 25 or higher.

The timeframe plays a vital role here because it contains the only information used for the computation. The AIS is also always calculated for a specific point in time because any new tweet by the Twitter user provides new data which in turn changes the attributable magnitude and in turn also the AIS.

We will now take a closer look at every step again and explain how the data is sourced, transformed and interpreted, as well as the exact calculations performed when computing the AIS.

6.2 Configuration

Before any data is fetched or prepared, the user must specify certain parameters. These parameters are necessary for data fetching as well as dataframe creation and they influence the AIS. Figure 9 shows the mask of the client application where the user enters these parameters.

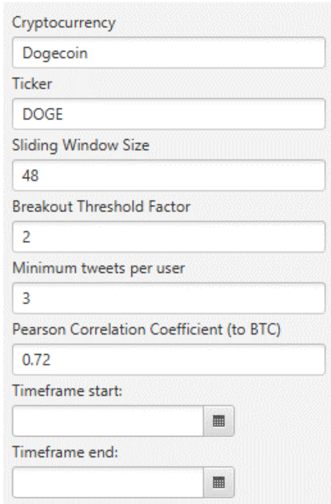

Figure 9: Example configuration of the AIS client

First, the user specifies the *cryptocurrency*. Whatever name is entered here is then used to build the query for the Twitter search. The *ticker* is used to fetch price data, as all assets (similarly to stocks) are only referred to by their ticker. Assets in general are often just called by their ticker in normal conversation, which is why the ticker is also incorporated into the Twitter search query as well.

To recall, the *sliding window size* is the number of previous time segments for which the average velocity is computed (in this example 48 hours or 2 days), which is then compared to the velocity of the current candle. The *breakout threshold factor* defines by what order of magnitude the velocity of the current candle must exceed the average velocity of the sliding window in order to be considered an anomaly. The *minimum tweets per user* describes how many tweets must be issued by a Twitter user about the cryptocurrency before being considered for AIS calculation. By increasing this threshold, the accuracy of the AIS typically increases, but at the risk of potentially skipping important users that have not met the threshold yet. In our experiments we found that the increased accuracy does not outweigh the risk of skipping users, we therefore recommend 3, which is the minimum because of how certain user-specific variables are calculated (see later).

Another variable is the *Pearson Correlation Coefficient* (or *PCC*) of the cryptocurrency to Bitcoin (BTC). The PCC describes, how heavily the asset is correlated to Bitcoin. How exactly this plays a role has been discussed in the previous chapter.

Finally, the user can specify the timeframe for which date the AIS shall be computed. The AIS will be computed at the end of the timeframe. The user has full control over the timeframe, but after analyzing market moves and experimenting with different start dates, we found January 1, 2020, to work very well.

6.3 Data Collection

6.3.1 Price data

Price data for both Bitcoin and the cryptocurrency being analyzed is fetched for the timeframe provided by the user. It forms the base for calculating the magnitudes of price changes and populating the dataframes. It is an integral part of the AIS, as we use it to derive how the market values and interprets Twitter activity, thus allowing for the subsequent calculation of the Asset Influence Score.

Price data is fetched from the Cryptocompare API[13], a free API which provides historic price data for a lot of trading pairs and exchanges. The data is fetched as OHLCV (*Open, High, Low, Close, Volume*) data for the most popular trading pair against the US-Dollar, preferably from the Binance[14] exchange.

OHLCV data, also commonly referred to as candlesticks, describes the price movement within a given timeframe. Trading volume is an indicator about how much buying & selling occurred in each time segment. For example, a high percentage price change with low volume is of less significance than one with high volume. Low volume means that less capital was required to make the price change and typically that fewer participants were involved.

An example of OHLCV data can be seen in Figure 10. *Open* and *Close* represent the ends of the body of each candlestick (the thicker part). So, on our hourly timeframe, *Open* is the price at the beginning of the hour, whereas the *Close* is the price 59 minutes 59 seconds into the hour. If the close price is above the open price, the candle is green, if it's below the open price, the candle is red. The data also contains a UNIX-timestamp, noting the start of the timeframe. The slim parts of the candlestick, the so-called wicks, are intra-timeframe price movements and represent the *Low* and *High* points of the price within the timeframe, meaning that the price had been there at one point within the hour. Large wicks are usually formed when the market overreacts to something, typically to positive or negative news. The volume is represented separately in the bar below the candle, a larger bar naturally indicating higher volume.

Figure 10: OHLCV data explained with a snippet of Dogecoin OHLCV on a daily timeframe; shown in Tradingview

[13] https://min-api.cryptocompare.com/documentation [accessed July 2022]
[14] https://www.binance.com/en [accessed July 2022]

The client application then checks if historic price data for the asset has already been fetched and can be found locally. If that is not the case, a call to the Cryptocompare API is made and the data returned by the API is cached on the user's machine. Since this data is historical and does not change over time, we can hereby save on bandwidth and improve performance significantly. Figure 11 shows an example of how such OHLCV data is stored for the timestamp 1633734000 (October 8, 2021, 23:00). *Volumefrom* represents the amount of Dogecoin traded, *Volumeto* the amount of US-Dollars.


```
{
  "time": 1633734000,
  "high": 0.245,
  "low": 0.2423,
  "open": 0.245,
  "close": 0.2441,
  "volumefrom": 3.2523777E7,
  "volumeto": 7916534.23
}
```


Figure 11: Example of OHLCV data

6.3.2 Tweets

Tweets form the counterpart to the price data and are used to derive what might've caused the price move. Collecting tweets is an integral part for computing the AIS, as it allows us to map and analyze price performance in relation to the activity of Twitter users. The data collection is performed via the Twitter API v2[15]. We have applied for and have been granted academic access[16] to the API, allowing us to query tweets for a specified timeframe and fetch historic data. Without academic access, we would've only been able to search tweets from the last 2 weeks. This is extremely important for us, as the period of highest volatility took place in the first half of 2021 – a time period which's tweets are only accessible through academic or paid access.

Tweets will be searched for via a query constructed from both the ticker-symbol and the full name of the cryptocurrency. The resulting Twitter-query for Dogecoin would be "Dogecoin OR DOGE". This query also incorporates all hashtags, like "#Dogecoin" or "#Doge". Additional data about the tweets will also be collected, those data points being the user ID, the unique tweet ID, the time stamp and the social metrics about the tweet (likes, retweets, comments, favorites). Only tweets written in English were considered.

Furthermore, the tweets were grouped into hourly time segments and stored in separate files for each hour. This allowed for convenient pairing to price action data, as the price data is also represented on an hourly basis.

The tweets for each timeframe were selected based on Twitter's selection for relevancy. We decided to limit the number of tweets fetched for each hour to 100. We noticed a very significant decrease in tweet quality and relevancy beyond the first 100 tweets (tweets from Bots or tweets that received next to no engagement),

[15] https://developer.twitter.com/en/docs/twitter-api

[16] https://developer.twitter.com/en/docs/twitter-api/getting-started/about-twitter-api#v2-access-level

therefore fetching more tweets would've only resulted in noisy data. Due to the sheer volume of tweets that exist, relying on custom sorting was not feasible due to the rate limitations posed by the API. The Twitter API only allows a user to fetch a maximum of 10 million tweets per month. Based on our selected time frame of 2 years (01.01. 2020 – 31.12.2021), this resulted in a dataset of roughly 1.5 million tweets for each cryptocurrency. We say roughly, because not every hourly segment contains 100 tweets about Dogecoin. Given the monthly limitation, we deemed this dataset size to be reasonable to ensure that we had enough headroom to experiment with different cryptocurrencies and fetch results multiple times if necessary.

Finally, due to the API's rate limits, it already takes the client roughly 15 hours to fetch the tweets for the chosen timeframe. Full-archive search is limited to 300 requests - so 300 hours within a 15-minute window. This is an extremely crippling constraint, but we could not find a way to expand the limit or find a workaround. Twitter does provide a tool to download data in bulk, but there is no way to limit the number of results. For Dogecoin alone, the dataset would contain roughly 47 million tweets, which would consume our API allowance for 5 months.

As mentioned, due to the significant decrease in tweet quality, it would actually be detrimental to fetch more than 100 tweets for any hour. It would result in huge, diluted data sets, only compromising output quality and client performance. Twitter's relevancy-first querying does a really good job at returning the tweets with the highest engagement. We experimented with fetching all tweets for a time segment and implementing custom sorting but were only able to match Twitter's relevancy-first results while using significantly more API-resources, further cementing our decision.

To interact with the Twitter API programmatically, we built a custom client using Retrofit[17]. We did not use pre-existing implementations like Twitter4j[18] because all we really needed was a lightweight implementation to fetch tweets via HTTP-GET. Using our own client was more performant, as it was built to only fetch data. It also removed the need for an external dependency. This also gave us full control of how the data is represented. In Figure 12 you can see an example of a tweet (trimmed to show the most relevant information).


```json
{
  "user": { "username": "EL_Know",… },
  "id": "1331108386600054784",
  "conversation_id": "1331108386600054784",
  "text": "#Dogecoin moving on up",
  "author_id": "199511040",
  "created_at": "2020-11-24T05:32:24.000Z",
  "lang": "en",
  "in_reply_to_user_id": null,
  "public_metrics": {
    "like_count": 8,
    "reply_count": 1,
    "retweet_count": 2,
    "quote_count": 0
  }
}
```


Figure 12: An example of a tweet fetched by our custom client

[17] https://square.github.io/retrofit/ [accessed June 2022]
[18] https://twitter4j.org/en/index.html [accessed June 2022]

The *user* object contains information about the author, like their username, the number of followers they have and when the account was created. The *id* is the id of the tweet assigned by Twitter. The *text* is the actual content of the tweet. *Author_id* corresponds to the user that issued the tweet. *Created_at* is the date and time when the tweet was issued. The *lang* is the language (English), which was a query criterion. *In_reply_to_user_id* would be set to a *user_id* if this tweet was issued as a reply rather than a standalone tweet. The *public_metrics* contain information about how many likes, retweets, replies and quotes the tweets got (engagement).

Local storage & caching

Knowing the limitations of the Twitter API, it became apparent very quickly that we needed a solution to store results locally to prevent repeated calls to the API. The grouped tweets follow a strict, uniform naming scheme and are stored as JSON-files locally.

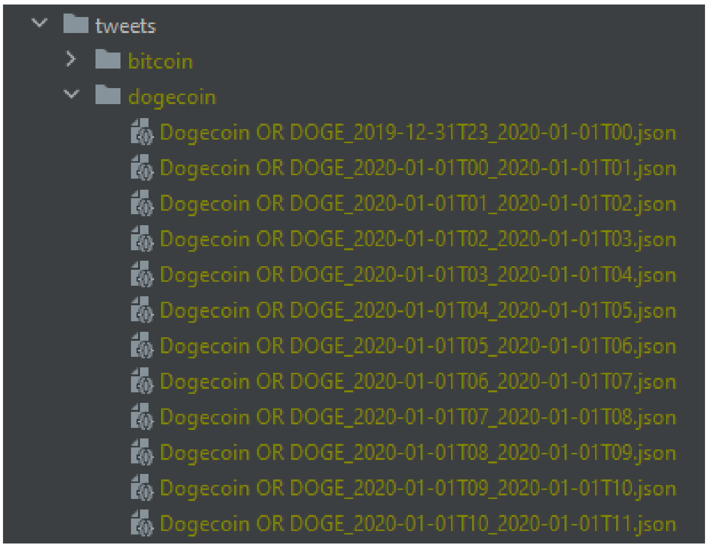

Figure 13: An example of the file structure for local storage of API requests

The data fetching client checks whether a corresponding file for a 1-hour timeframe is present and only issues a request to the API if that is not the case. How these files are stored can be seen in Figure 13. Otherwise, it instead reads the content from the respective file and returns that instead. If a request is issued, the result is immediately stored in a file. Since the data is historical and not time sensitive, the results would be the same every time. By caching tweets, we were able to reduce execution time from roughly 15 hours down to roughly 3 minutes (after the first execution of course), so this was without a doubt mandatory. Also, if we would not cache the tweets, we could only execute the AIS calculation a couple of times per month before running out of API-calls.

6.4 Data transformation and representation as Dataframes

We now have two separate, locally cached lists of files, those being price data and tweets. Both are already grouped into hourly segments. To streamline and standardize the computation process for calculating the AIS, we proceeded to aggregate and fuse the data into what we refer to as dataframes.

A dataframe represents an hour of market and twitter activity and consists of the hourly OHLCV data, the (up to 100) most popular tweet from any given hour as well as some statistics which are computed for every dataframe. Before calculating the AIS's for the twitter users contained in the data set, the client creates these dataframes for the desired period. In Figure 14 you can see an example of such a dataframe. We want to note that the tweets contain a lot more fields and have been omitted for this example. How tweets are represented in a dataframe will be discussed shortly.

```
{
  "ohlc": {
    "time": 1621371600,
    "high": 0.4805,
    "low": 0.4733,
    "open": 0.4767,
    "close": 0.4748,
    "volumefrom": 5.75389169E7,
    "volumeto": 2.745839092E7
  },
  "ohlc_statistics": {
    "window_size": 24.0,
    "index": 4575,
    "mean_velocity": 79.705291907675,
    "mean_volume": 6.140638243791666E7,
    "previous close": 0.4767
  },
  "tweets": [
        {"id": "1394772833608224769"},
        {"id": "1394774065378963457"},
        {"id": "1394772694768525312"},
        …
   ]
}
```

Figure 14: An example of a dataframe

The OHLCV data for every hour is directly imported from the price data and remains unchanged. We also add another a field called `ohlc_statistics`. The values are computed for each dataframe, based on the previously described sliding window approach. The client takes the previous {`window_size`} number of ohlc data points and computes the *mean velocity* of the sliding window. Our dataframe example has the index 4575. Therefore, the mean velocity is comprised of the velocities of the dataframes 4550 – 4574, because our window size is 24.

We have introduced the *velocity* as a metric that describes the trading activity in each timeframe. It is composed of both price movement and volume (taken from the ohlc-section of each dataframe) and is calculated as follows:

$$Velocity_{ohlc} = \frac{(High_{ohlc} - Low_{ohlc}) * VolumeTo_{ohlc}}{10000}$$

We use the high and low instead of the open and close to capture excessive price movements that might've corrected themselves within the timeframe. We also divide the Velocity by 10.000 because the number can become very large quite quickly, this just makes it more useable. We experimented with different ways to lower the velocity to make it easier to work with and found a simple division by 10.000 fitting.

The mean velocity over the past `{window_size}` candles is calculated as follows:

$$MeanVelocity_{window} = \sum_{i=0}^{window\_size} \frac{Velocity_{ohlc_i}}{window\_size}$$

It is important to note that index `i` in this case refers to the index within the sliding window and not within all dataframes. After a dataframe is created, the sliding window moves to incorporate the new dataframe and in turn drops the dataframe at prior index 0.

We also store the *mean volume* from the sliding window, which is computed analogously to the *mean velocity* using the *volumeto* from the ohlc-segments in the previous dataframes. Both the mean volume and the mean velocity are stored as BigDecimals to avoid rounding errors. Finally, we store the close price of the previous dataframe's ohlc, so we can compute percentage price moves that have occurred between dataframes.

The last segment of a dataframe a collection of the top 100 tweets of that hour that got the most engagement. The original tweets are slightly trimmed to only contain relevant information which is used either for displaying in the client application or incorporated in the AIS calculation. An example of such a tweet can be seen in Figure 15.


```
{
  "user": {
    "username": "elonmusk",
    "name": "Elon Musk"
  },
  "id": "1377567762919292938",
  "text": "SpaceX is going to put a literal Dogecoin on the literal moon",
  "created_at": "2021-04-01T10:25:23.000Z",
  "public_metrics": {
    "like_count": 519100,
    "reply_count": 24426,
    "retweet_count": 49631
  },
  "is_original": true,
  "associated_outbreak_magnitude": 1721.1555434957272
}
```


Figure 15: An example of how a tweet is stored in a dataframe

We start with the user-object, which contains a *username* (must be unique within Twitter) and the *name* displayed for the user. The tweet *id* is a unique ID given to every tweet by Twitter. The *text* contains the content of the tweet, *created_at* contains the date and time the tweet was issued. These fields remain unchanged from the API result, so do the *public_metrics*. *Is_original* is a field which was added by us and states, whether this tweet was issued as a reply to another tweet or as a standalone tweet. We only want to use original tweets for calculating the AIS, because they typically garner more engagement and are more representative of how a user's followers value and interpret their tweets. Lastly, we store the *associated_outbreak_magnitude*, representing the change in magnitude adjusted for normal fluctuation and Bitcoins magnitude. The *associated_outbreak_magnitude* is the same for every tweet in the dataframe and then weighted according to the engagement share.

Now that we have a set of dataframes for the timeframe, we can move on to the next step, which is the computation of the AIS.

6.5 Gaining Insight

In this chapter we describe the math and the procedure involved when computing the AIS. We define and perform a series of calculations to extract information and identify correlation between twitter activity and price action.

6.5.1 Prerequisites

We first want to restate the importance of the definition of *Magnitude (M)*. *M* describes the factor by which the velocity of a time segment differs compared to average velocity of the sliding window.

$$M_{ohlc} = \frac{Velocity_{ohlc}}{MeanFluctuation_{sliding\_window}}$$

If for example a time segment has a magnitude of 10, this signals a ten-fold increase in trading velocity in this hourly segment compared to the average velocity. To recall, velocity is defined as follows:

$$Velocity_{ohlc} = \frac{(High_{ohlc} - Low_{ohlc}) * VolumeTo_{ohlc}}{10000}$$

All the data points required to calculate these variables are contained within the dataframe. They are written as $DataPoint_{section}$, *DataPoint* representing the actual field (for example *High*), whereas the *section* represents the section of the dataframe where this field is found (for example *ohlc*). *DataPoints* can also represent variables which have to be calculated first, as for example the *Velocity* in the *Magnitude* calculation represents the result of calculating the *Velocity* for a respective *ohlc*-segment in a dataframe.

The *Breakout Threshold Factor (BTF)* also plays an integral role during the AIS. It is a **user-chosen parameter** and is not calculated. The BTF defines the threshold which a segments magnitude needs to exceed to be considered an anomaly. The goal of the AIS is to describe the probability of a users' tweets occurring within an anomaly period, so the BTF sets the bar for the magnitude. If the BTF is set too high this will result in many low AIS-scores, which can increase meaningfulness of high AIS-scores at the tradeoff of not getting any meaningful results at all. If a user can score a high AIS for a high BTF, the certainty of this users' tweets being weighted heavily by the market increases significantly. If, on the contrary, the BTF is set too low (for example 1.05), many users will score very highly, making influential users indistinguishable from non-influential ones. It is therefore the user's obligation to experiment with different BTF's to achieve results which means their expectation. Generally, more volatile cryptocurrencies require higher BTF's to see meaningful results, compared to less volatile assets. With the BTF defined, we can now define an anomaly period as a time segment which's magnitude exceeds the BTF. Again, our goal is to find users which's tweets tend to occur within such anomaly periods.

Lastly, we need a measure that differentiates the potential impact of twitter users from one another. Here, we use the *Engagement (E)* a tweet generates, and subsequently the *Engagement Share (ES)* of the tweet in its respective time period. We define *Engagement* as follows:

$$E_{Tweet} = Likes_{Tweet} + 2 * Retweets_{Tweet}$$

We choose to weight retweets twice as much as likes because retweeting requires more effort and typically represents a stronger positive reception of the content. We experimented with different weightings of the retweets and found 2 to be the sweet spot which maximizes the meaningfulness of this metric.

The *Engagement Share (ES)* is the fraction of engagement of the respective tweet in relation to the sum of all engagement over all tweets that occurred within the respective time segment `t` and is defined as follows:

$$ES_{Tweet} = \frac{E_{Tweet}}{\sum_t E_{Tweet}}$$

To capture the different market dynamics at play, we extended the Magnitude calculation by incorporating the respective magnitude of Bitcoin as well as the assets correlation to Bitcoin using the *Pearson Correlation Coefficient (PCC)*. In the current state of the market, almost all crypto assets correlate and move in tandem with Bitcoin, therefore this must be considered. We call this extended metric *Magnitude attributable to External Factors (MEF),* suggesting that all natural price movement options have already been taken into consideration.

$$MEF_{ohlc} = M_{ohlc} - (M_{bitcoin} * PCC)$$

The Magnitude of Bitcoin is calculated in an equivalent manner to the analyzed asset (see Magnitude) and shall only be deducted when the asset for which the calculation is being performed is not Bitcoin.

6.5.2 TweetMap

To estimate the influence of a Twitter user over the asset in question we utilize the created dataframes. We iterate over the dataframes between a predefined `start` and `end` time, creating statistics for each user that has tweeted within the specified period. These mappings are stored in a temporary data structure called a `TweetMap`. TweetMaps essentially represent the knowledge gained about a single user spanning the chosen time period. These TweetMaps contain the following fields about a user:

```
User user;
Int anomalyTweetCount;
Int regularTweetCount;
Double totalMagnitude;
List<Tweet> tweets;
```

The algorithm for creating and populating these `TweetMaps` is rather simple and works follows (represented as pseudocode):

```
HashMap<String, TweetMap> map; // <username, tweetmap>

For all Dataframes from start to end:
      Double engagementSum = 0;

      For all tweets in current dataframe:
            engagementSum += engagement(tweet)

      // iterate a second time because we need engagementSum
      For all tweets in current dataframe:
            if (tweetMap for user doesn't exist):
                   Create tweetMap, set user and add to map
            if (tweet is anomaly):
                  tweetMap.incrementAnomalyCount()
            else:
                  tweetMap.incrementRegularCount()
            tweetMap.addToMagnitude(MEF(tweet))
            tweetMap.addTweet(tweet)
```

After the mapping process is completed, we are left with a snapshot of the current user activities. We can then leverage this knowledge to determine the users which's tweets are most frequently involved in abnormal trading activity.

6.5.3 Calculating the AIS

The AIS describes the probability that on a given day, a specific user's newly issued tweet will co-occur in an anomaly period.

For every user, the associated TweetMap contains a list of all tweets that the user has issued. These tweets in turn contain information about the MEF for the respective hourly period, as well as the information necessary to calculate engagement. All the following steps are taken on a **per-Twitter-user basis**.

We start by sorting all the users' tweets according to their associated *Magnitude Attributable to External Factors (MEF*. We then take the tweets at the middle 3 indices (middle, middle +1, middle -1) and use these tweets to calculate a *Median Magnitude (MM)* for the user. The *MM* is the average MEF of those middle 3 tweets. The Median Magnitude allows us to accurately estimate the average Magnitude of the time periods during which tweets from a specific user occurred.

$$MM_{user} = \frac{MEF_{T_{middle-1}} + MEF_{T_{middle}} + MEF_{T_{middle+1}}}{3}$$

In Table 5 you can see a list of all tweets issued by Elon Musk between January 1, 2020, and December 31, 2021, which matched the Twitter query "Dogecoin OR Doge". The size of the sliding window has been set to 24. These Tweets have already been sorted by their MEF.

Rank	Text	Likes	Retweets	ES	MEF
1	SpaceX is going to put a literal Dogecoin on the literal moon	519100	49631	98,46	1.001,54
2	Tesla will make some merch buyable with Doge & see how it goes	537710	53826	88,11	661,304
3	One word: Doge	211995	24555	99,70	315,917
4	No highs, no lows, only Doge	731126	104380	55,59	75,708
5	Dogecoin is the people's crypto	524540	95690	42,34	75,708
6	Doge day afternoon	165895	17330	85,30	33,730
7	Doge spelled backwards is Egod	387003	35443	99,03	26,719
8	Release the Doge!	126205	15455	95,63	22,480
9	Bought some Dogecoin for lil X, so he can be a toddler hodler	518767	54402	95,15	18,600
10	If major Dogecoin holders sell most of their coins, it will get …	312013	32059	98,13	16,512
11	Doge meme shield (legendary item)	269620	24881	97,47	13,120
12	Baby Doge, doo, doo, doo, Baby Doge, doo, doo, doo …	289658	58,97	97,71	13,018
13	Who let the Doge out	744147	105373	93,72	12,085
14	Working with Doge devs to improve system transaction efficiency.	524899	80119	93,53	9,493
15	Do you want Tesla to accept Doge?	386818	92668	92,50	5,813
16	If you'd like to help to develop Doge, please submit your ideas …	185050	30930	91,24	4,915
17	How much is that Doge in the window?	298090	51330	92,83	2,609
18	SpaceX launching satellite Doge-1 to the moon next year. …	518387	109441	93,27	2,510
19	Doge Barking at the Moon	318313	46476	96,84	2,174

Table 5: All tweets issued by Elon Musk about Dogecoin between January 1, 2020, and December 31, 2021

We have chosen the approach of taking the middle 3 tweets for a few reasons. Firstly, this allows us to incorporate users with a minimum number of tweets of 3. Going any higher would mean that users must issue increasingly more tweets before being eligible for AIS calculation. Our experiments have furthermore shown that incorporating more tweets does not increase meaningfulness of the *MM*, but rather decreases it. If we for example took the average of all MEF's for the tweets issued by Elon Musk, we would have an average MM of 121.79. As you can see in Table 5, only 3 out of 19 tweets exceed this MM, while all the other tweets lie significantly below that number.

We can now calculate the MM for Elon Musk:

$$MM_{elonmusk} = \frac{18{,}60 + 16{,}512 + 13{,}12}{3} = 16{,}077$$

This figure gives a significantly more accurate estimate of what average MEF a tweet by Elon Musk co-occurs with.

We use the same approach to calculate the *Median Engagement Share (MES).* We introduced the MES to approximate the average engagement share a user can expect from an original tweet which involves the cryptocurrency. We do not sort according to engagement share, because we want the MES to be tied to the median tweets. The tweets considered are the same which are taken to calculate the MM.

$$MES_{user} = \frac{ES_{T_{middle-1}} + ES_{T_{middle}} + ES_{T_{middle+1}}}{3}$$

Applying this formula to our example, the MES for Elon Musk is calculated as follows:

$$MES_{elonmusk} = \frac{95{,}15 + 98{,}13 + 97{,}47}{3} = 96{,}91$$

By combining the MM and the MES, we can then calculate the *Median Attributable Magnitude (MAM),* which forms the basis for calculating the AIS. We introduced the *MAM* to represent the average change in trading magnitude which can be attributable to a particular user. ***If user x tweets, we can expect a magnitude for the respective time segment of the Median Attributable Magnitude of user x.***

$$MAM_{user} = MM_{user} * \frac{MES_{user}}{100}$$

For Elon Musk, this means his Median Attributable Magnitude is calculated as:

$$MAM_{elonmusk} = 16{,}077 * \frac{96{,}91}{100} = 15{,}58$$

We have now introduced and defined all variables required to compute the AIS for a user. To recall, the AIS is defined as the certainty of a user's newly issued tweet occurring within an anomaly period. An anomaly period is defined as the Magnitude of a time segment exceeding the (user-chosen) Breakout Threshold Factor (BTF). This leads to the following definition of the AIS:

$$AIS = \frac{MAM_{user} * 100}{BTF}$$

For our first example we chose a BTF of 25.

$$AIS_{elonmusk} = \frac{15{,}61 * 100}{25} = 62{,}32$$

Put into words, this means that given the timeframe from January 1, 2020, until December 31, 2021, Elon Musk's AIS with a BTF of 25 on December 31, 2021, for Dogecoin is 62,32. This means that a newly issued tweet from Elon Musk that is related to Dogecoin has a 62,32%-likelihood of co-occurring within an hour where Dogecoin-trading will experience a magnitude of 25 or higher.

Let's look at another example to see what impact a different BTF has on the AIS:

$$AIS_{elonmusk} = \frac{15{,}58 * 100}{50} = 31{,}16$$

If we set the BTF to 50, the AIS of Elon Musk shrinks to 31,16, meaning the certainty of a newly issued tweet by Elon Musk about Dogecoin occurring within a time segment with a magnitude of 50 or greater sinks to 31,16%.

In this state, it is possible for a user to exceed an AIS of 100. Because we have introduced the AIS as a certainty-score, we extended the AIS by an additional rule. If the AIS of a user exceeds 100, we instead use the users *Anomaly Ratio (AR)* as their AIS. A users' AR is defined as follows:

$$AR_{user} = \frac{number\_of(Tweets_{anomaly})}{number\_of(Tweets_{total})}$$

The anomaly ratio describes the percentage of a users' tweets occurring within an anomaly period.

This rule applies if we calculate the AIS for Elon Musk with a BTF of 10:

$$AIS_{elonmusk} = \frac{15{,}58 * 100}{10} = 155{,}80$$

In this case, the AIS would exceed the threshold of 100. We therefore apply the rule of using the anomaly ratio of Elon Musk on December 31, 2021, which is 68.42%. This can easily be calculated by looking at Table 5. We divide the number of tweets with a magnitude of 10 or above (13) by the number of total tweets (19).

For a BT of 10, Elon Musk would therefore have an AIS of 68.42, which translates to the certainty of a newly issued tweet occurring within a period with magnitude 10 or higher of 68,42%.

This means that an AIS of 100 can only be achieved if every tweet by a user has occurred within a time segment which's magnitude has exceeded the BTF.

Choosing an appropriate BTF depends on the users' goals. If the user wants to only identify users with a very high probability of having a certain degree of influence over the market, we recommend setting the BTF higher. If the user wants to identify a set of Twitter users with at least some suggested influence over the market, we recommend setting the BTF lower. In general, we recommend trying a few different BTF-configurations. If one or more users manage to achieve an AIS of 100, we recommend lowering the BTF. If not a single user manages to achieve an AIS above 10, the BTF might be set too low. In general, we can't issue a recommended BTF, as it is heavily dependent on the cryptocurrency.

How exactly we evaluated the AIS in a real-world setting will be discussed in the next section of this paper.

6.6 Experiments & Evaluation

Since inception, Dogecoin, at the time of writing (September 2022), is up roughly 71000%, sitting at roughly 0,06 dollars, up from its all-time-low of 0,000087 dollars. Capturing these % gains as an investor is extremely unlikely, given the fact that the currency was pretty much unheard of in its beginnings, and it is unrealistic to assume that a rational investor with a standard risk tolerance would enter a trading position at this time. The likelihood of the investor never having heard of Dogecoin is, as mentioned, also very high.

In this section we discuss and evaluate the AIS in a simple trading environment. Our aim is to find the best performing configuration and compare it to simple buy-and-hold strategies to determine whether it would've been possible to out-trade the market. We compare it to a buy-and-hold strategy as it a very common, hands-off investment strategy practiced by many individuals and institutions. It relies on achieving historic market returns instead of actively managing positions. All tests will be performed on Dogecoin, a cryptocurrency that had experienced a meteoric rise in price, closely tied to the Twitter activity of Elon Musk.

We believe it is realistic to assume that an average, decently crypto-savvy investor could've started investing in Dogecoin after its first strong appearance in mainstream media at the beginning of 2021 after it had reached a price of 0,01 dollars, which was around the time of January 5, 2021. In Figure 16 you can see the price performance of Dogecoin, starting in January 2021 and ending in the middle of 2022.

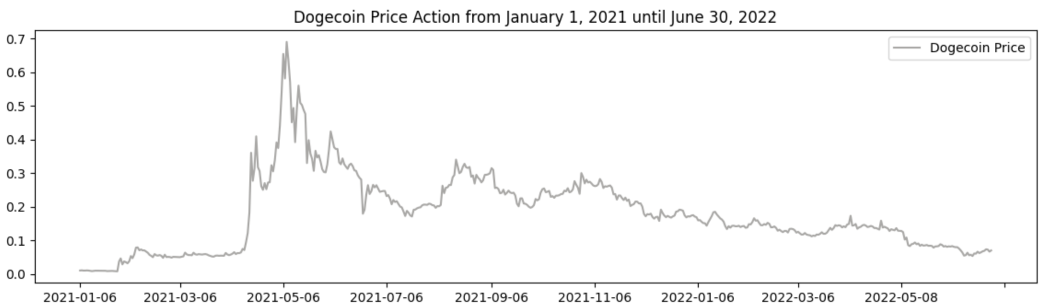

Figure 16: Dogecoin Price Action from January 1, 2021, until June 30, 2022

The price chart of Figure 16 coincides with the buy-and-hold period that will serve as our baseline. The period featured a very bullish start, reaching a top of roughly 70 cents, followed by a slow and rather steady decline in price.

We test whether it would've been possible to use the AIS as a buying indicator to capture more financial gains over the analyzed period, compared to the buy-and-hold strategy. The simulated traders will hold their positions for varying durations. The investor and the AIS traders will be equipped with the same starting balance at the beginning of the analyzed period, and we will compare their portfolio performance throughout the period to determine, whether the traders would've been able to outperform the buy-and-hold investor and by how much.

Figure 17 shows an example performance of the buy-and-hold investor, which serves as a baseline comparison. The investor was given 10,000$ to start.

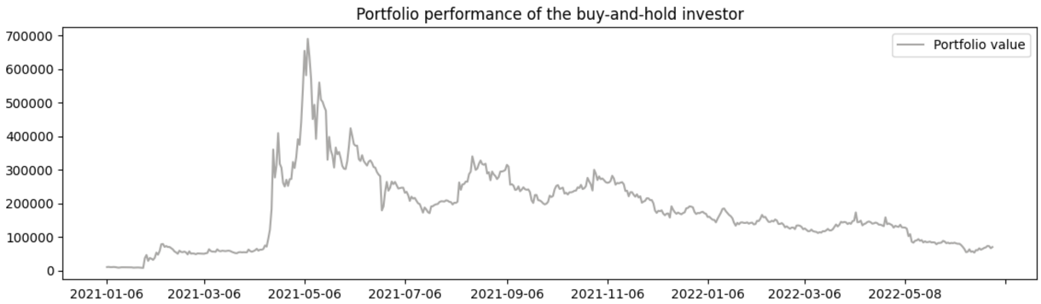

Figure 17: Portfolio performance of the buy-and-hold investor

As you can see, the portfolio performance perfectly correlates to the price action of Dogecoin shown in Figure 17. The investor managed to achieve a peak portfolio value of roughly 700,000$ in early May 2021. On December 31, the investor portfolio was worth 171,000$. On June 30, 2022, the investor portfolio was worth 69,560$.
We will now go over the trading algorithm which employs the AIS as a buy-signal, followed by some optimizations which were performed to increase trading performance. Finally, we will compare the results to the buy-and-hold portfolio.

6.6.1 Trading algorithm

To test the effectiveness of the AIS as a trading indicator, we wrote a simple algorithm that executes virtual trades using only the AIS as a buy-signal and tracks portfolio performance accordingly. We use Dogecoin as our traded currency.

Trade performance is tracked in a simple data structure called Trader:

```
class Trader {
      final Long holdDuration;
      double balance = 10000;
      double currentHoldings = 0;
      double latestEntryTimestamp;
}
```

Each Trader is equipped with a starting balance of 10000, which represents a dollar value. Whenever a Trader enters a position, the amount is deducted from his `balance` and the `currentHoldings` increase to the number of Dogecoin bought. For example, if a Trader buys 200$ worth of Dogecoin at 5 cents, his balance changes to 9800 and his `currentHoldings` change to 4000 (Dogecoin). This exposure allows the Trader to profit or experience losses from price changes in Dogecoin, while anything in his balance is considered stable.

For each Trader a different `holdDuration` is specified. The `holdDuration` is specified in days. When the `holdDuration` is 0, the Trader sells all his positions at the end of each day. When the `holdDuration` is 1 Day, the Trader sells all his positions at the end of the following day and so on. The `latestEntryTimestamp` tracks the entry for the latest position. When the current timestamp exceeds the `latestEntryTimestamp` + the Traders `holdDuration`, the position is closed and any profit or loss is realized. If a buy-signal is issued while a Trader is currently holding a position, the position is kept or expanded and the `latestEntryTimestamp` is renewed.

A trader will enter a position based on his available balance and the AIS of the user that tweeted. If for example a user with an AIS of 15 issues a tweet, the user will enter the trade with 15% of his currently available balance. We will only consider the tweets from the top 8 users ranked by AIS as buy-signals.

The trading algorithm works as follows (represented as pseudocode for convenience):

```
trade (start, end) {
   dailyTimestamp = start
   hourlyTimestamp = start

   while (dailyTimestamp < end) {

      tweetMaps = map with available data until dailyTimestamp

      while (hourlyTimestamp < dailyTimestamp + length(day)) {
         for (user : top 8 users by AIS with > min tweets from tweetMaps) {
            for (tweet : tweets from dataframe (currentHourlyTimestamp)) {
               if (user.username == tweet.username) {
                  // high AIS tweet - enter a position with all Traders
                  availableBudget = trader.balance * user.AIS / 100
                  entryPrice = open + (open - close) / 5 // 20% premium
                  entryAmount = availableBudget / entryPrice
                  trader.currentHoldings = entryAmount
                  trader.balance -= availableBudget
                  trader.latestEntry = dailyTimestamp
               }
            }
         }
         hourlyTimestamp += duration(hour)
      }
      // end of day, close open positions
      if (trader.currentHoldings > 0) {
         if (dailyTimestamp >= trader.latestEntry + trader.holdDuration) {
            tradeResult = t.currentHoldings *
                        dataframe(hourlyTimestamp).openPrice
            trader.balance += tradeResult
            trader.currentHoldings = 0
         }
      }
   }
}
```

To make things more realistic we decided to add a small premium to the entry price. At any given hour, a trader can only capture 80% of the price move. This is to emulate that there are most likely already bots at work that automatically buy upon the tweets of certain users. Therefore, even the fastest trader will not get the best possible price.

6.6.2 Trading strategy

To recall, the traders in the trading algorithm purchase Dogecoin whenever one of the top 8 users ranked by AIS issues an original tweet. The simulated traders differ in terms of how long they hold their positions. We will compare different holding durations of the traders to determine the most optimal one. The traders always sell at the end (23:59 o'clock) of the last day of their holding period. We will test a end-of-day, 1-day, 2-day, 3-day, 1-week and 2-week holding period for the traders. If a new tweet by one of the top 8 users is issued, the holding period is renewed. For example, if a new tweet is issued after one week, the 2-week trader will be holding a total of 3 weeks, whereas the end-of-day trader will enter a position only for the day that the tweet was issued. These durations have been chosen to cover different timeframes, while also still being considered trading periods. Experiments have shown that by exceeding a 2-week holding period, we are essentially matching the buy-and-hold investor, as the likelihood of a new tweet being issued within 2 weeks (and thus renewing the holding period) is rather high.

The period we will be testing starts on January 5, 2021. This date coincides with the date where we believe a crypto-savvy investor could've made their investment. The period will end on December 31, 2021. First, we will work out the optimal parameters for the AIS for our trading period. This will be done by comparing different parameter-combinations and maximizing the balance at the end of the duration. We will then compare our results to the results of the buy-and-hold investor to determine whether an AIS trader was able to outperform the investor.

6.6.3 Optimizing Parameters

Here we experiment with different user-configurations of the AIS client. As already discussed in Chapter "Configuration", we can tweak some variables in order to maximize the traders balance at the end of the analyzed period.

Sliding Window Size

We first evaluate the optimal size of the sliding window. To recall, the sliding window size defines over how many previous hours the average velocity is calculated. As the average is calculated during dataframe creation, we have to re-create the dataframes for every window size. We generated dataframes with window sizes 6, 12, 24, 48, 172 and 340. We set the BTF to 8, the minimum number of tweets to 5 and proceeded to run the trading algorithm on each set of dataframes.

The BTF and the minimum number of tweets were chosen somewhat arbitrarily but were frequently used during development and showed decent results. As the differentiating factor is only the size of the sliding window, this allows for adequate comparison regardless of the other parameters. As mentioned previously, each trader starts with a capital of 10,000$ on January 5th. In Table 6 you can see the end balances for each trader on December 31, 2021.

Size	End of Day	1 Day	2 Days	3 Days	1 Week	2 Weeks	Average
6	45387.56	36177.44	49407.40	38986.07	127588.14	154913.37	75410,00
12	67688.40	50889.20	67504.32	45935.49	252466.45	206455.67	115156,59
24	71758.59	52326.96	70671.55	46486.30	250723.63	205866.04	116305,51
48	74394.88	52736.8	71715.06	47494.74	250722.78	205865.34	117154,93
172	66534.83	49482.97	66271.30	45973.84	255118.37	208273.18	115275,75
340	69232.41	63102.69	83953.36	64836.76	79376.03	218136.53	96439,64
Avg	65832,78	50786,01	68253,83	48285,53	202665,90	199918,35	105957,07

Table 6: End-balances of the trading algorithms on December 31, 2021, trading Dogecoin with different window sizes. BTF = 8, minimum Tweets = 5

It becomes apparent that initially, the trading algorithm performs better with increased window size, the benefit of which reduces as we exceed a window size of 48, which achieved the best result with an average balance of 117,154.93$. We will therefore choose a window size of 48 (equivalent to two days) for the next experiments. We want to note here that none of the averages were able to come close to the results of the buy-and-hold investor, indicating that a BTF of 8 in combination with a minimum number of 5 tweets might not be an optimal configuration.

Generally, we also want to point out that regardless of the window size the traders were always able to generate a sizeable return. Since the buy-and-hold investor had managed to increase his balance from 10,000$ to over 170,000$ (a 1700% increase), this is simply due to the fact that the underlying asset – in our case Dogecoin – had performed exceptionally well. This also indicates that the configuration is in fact important, as we can achieve different results with different window sizes, further indicating that the AIS trading algorithm does require parameter optimization. Having found a good sliding window size, we now move on to adjusting the other parameters, namely the minimum number of tweets issued by the users and lastly the BTF.

Minimum number of tweets

After establishing a well-performing window size, we repeated the experiment, fixing the window size to 48 and the BTF to 8. This time, the variable parameter is the minimum number of tweets a user must have had issued in order to be considered as one of the top 8 users by the trading algorithm.

Min Tweets	End of Day	1 Day	2 Days	3 Days	1 Week	2 Weeks	Average
3	119852,91	63007,96	77967,21	28797,02	144459,53	249176,34	113876,83
4	130943,78	92160,02	42382,25	28663,85	68427,81	245187,49	101294,20
5	114019,22	108497,21	57762,37	31252,41	71900,51	231998,33	102571,68
8	99549,45	57406,23	78539,50	56047,39	213209,46	205323,73	118345,96
10	88046,93	37452,31	64394,85	92522,10	132235,42	202378,90	102838,42
15	41905,60	62429,73	43165,17	47655,76	46462,82	123392,35	60835,54
20	27666,09	32944,7	42839,58	72964,36	57663,81	129381,13	60576,62
30	26720,70	39849,60	3132,53	38053,41	69415,57	136404,34	52262,69
Avg	108117,48	82291,30	68363,91	65992,72	133962,49	253873,77	118766,94

Table 7: End balances of the trading algorithm on December 31, 2021, trading Dogecoin; With varying number of minimum Tweets. BTF = 8, Window Size = 48

It becomes apparent that, roughly speaking, trading performance decreases with an increase in the minimum tweets required. In this case, the iterations with minimum tweets above 19 did not even trade upon a single tweet issued by Elon Musk, as he has only ever tweeted about Dogecoin 19 times before December 31, 2021.

Generally, we think that setting this parameter as low as possible (which is 3) is most optimal, as the AIS heavily incorporates engagement, which is decided by the public perception of the user.

A user with 1000 tweets will still score lower in terms of AIS than a popular user with only 4 tweets. Even though the iteration with 8 as the minimum number of tweets did achieve the highest average end balance, we believe this to be an outlier that caused this iteration to miss out on a losing trade caused by somebody within the 3-7 tweet range.

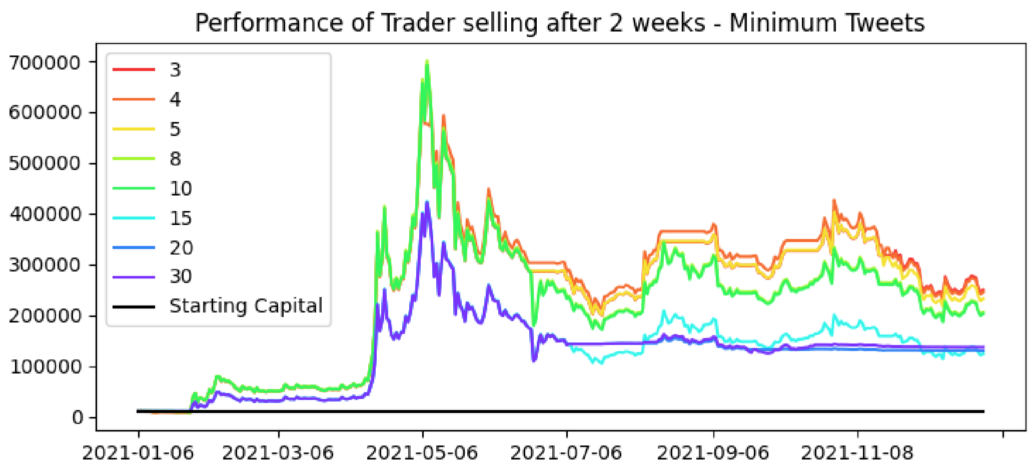

Figure 18: Results of the trading algorithm running for Dogecoin with varying number of tweets per user required to be considered as a buy-signal; selling after 2 weeks

In Figure 18, the relationship between the minimum number of tweets and the trading results becomes visible. In this case, a higher minimum number of tweets causes the algorithm to miss out on primary influencers (which tend to not tweet as often). Highly impactful tweets, like when Elon Musk tweets about Dogecoin, will then likely be tweeted about by users which match the minimum number of tweets, which therefore causes the algorithm to trade off the input of lower AIS individuals. This results in a smaller position size and therefore smaller capital gains.

An in our opinion very interesting find are the results from the trader that sells end of day. As you can see in Figure 19, the trader was able to generate 1200% returns with a minimum number of tweets set to 3 & 4. This is impressive because this result was achieved with relatively little market exposure, as the investor returned to a cash position at the end of each trading day.

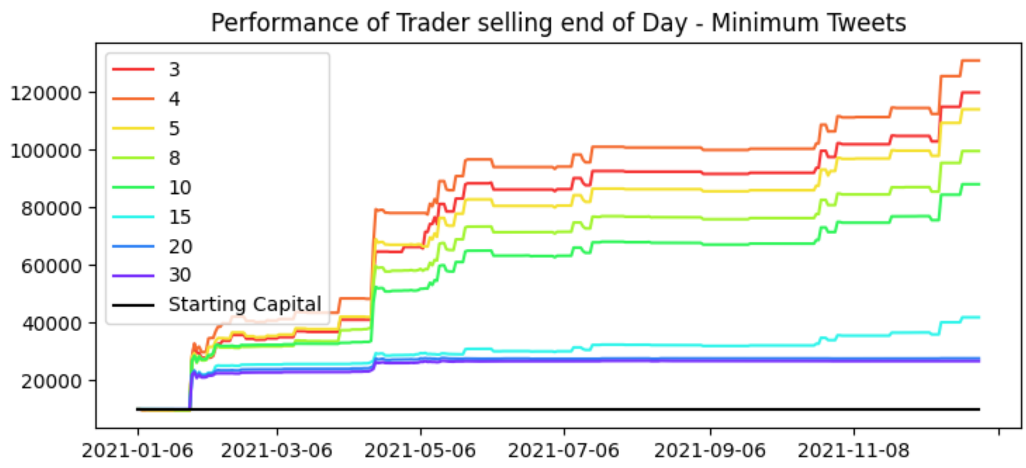

Figure 19: Results of the trading algorithm running for Dogecoin with varying number of tweets per user required to be considered as a buy-signal; selling at the end of each trading day

The trader that sells at the end of each day does overall deliver a worse performance than the trader that holds each position for at least two weeks, but at the benefit of having very little opportunity cost associated due to his liquidity during the trading period. The algorithm also experiences little losses as can be seen in Figure 19 and Figure 20, suggesting that the AIS might be a suitable indicator for positioning in markets influenced by Twitter.

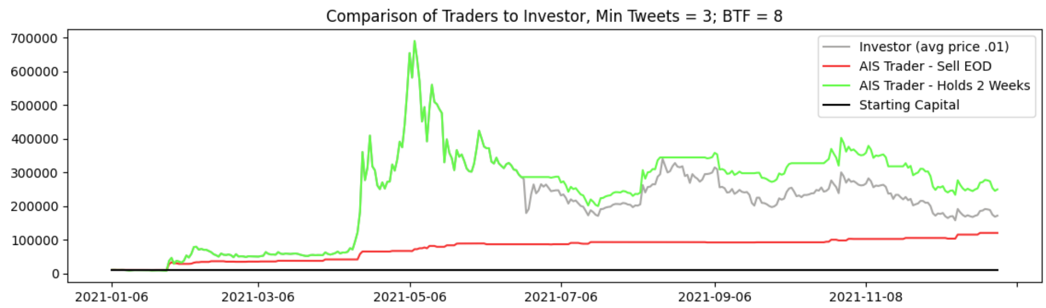

Figure 20: The results of the trading algorithm on December 31, 2021, trading Dogecoin; compared to an investor that bought at the same time as the trading algorithm started, minimum Tweets = 3, BTF = 8

Having established the best performing minimum number of tweets is 3, therefore we will continue with this value. To recall, the minimum number of tweets is 3 because we have introduced the Median Attributable Magnitude (MAM) as the average over 3 tweets. By making this the minimum, we can ensure that the MAM for every user is computed in the same manner. We can now move on the final parameter set by the user, which is the Breakout Threshold Factor (or BTF).

Breakout Threshold Factor

Similarly to the minimum number of tweets required, a strong negative correlation between the size of the breakout threshold and the average return becomes noticeable, which can be seen in Table 8. This can be explained through the heavily reduced number of trades the algorithm makes with an increased BTF, causing it to miss out on substantial returns during the time of a market uptrend.

BTF	End of Day	1 Day	2 Days	3 Days	1 Week	2 Weeks	Average
2	251122,60	169221,79	195723,66	50688,97	97240,69	239620,25	167269,66
5	170976,14	75286,92	82494,24	27409,07	133512,85	246785,50	122744,12
8	119852,91	63007,96	77967,21	28797,02	144459,53	249176,34	113876,83
10	106854,66	57475,04	72718,64	27973,69	148163,44	248085,47	110211,83
15	81090,10	5244,20	62312,01	24285,76	132857,75	243789,13	91596,49
20	68484,31	4432,59	53714,06	23719,60	133798,80	235010,62	86526,66
30	47696,88	36223,97	50389,70	24190,18	136063,94	219361,73	85654,40
50	29091,49	24463,60	34010,05	20565,56	101601,65	194278,54	67335,15
100	17962,3	16419,29	22106,45	16892,14	58464,62	150754,57	47099,90

Table 8: Results of the trading algorithm trading Dogecoin; started on Jan 5 2021, balances recorded on Dec 31, 2021

This could either indicate that a lower BTF might be better in general, even for highly volatile assets like Dogecoin, or that it is just more advantageous to frequently enter trades in bullish market conditions. To further validate this, we will extend the timeframe until June 2022. If you recall, Dogecoins meteoric rise was followed with a long period of slowly deteriorating price action.

6.6.4 Assessment of the AIS as a Trading Indicator

As demonstrated during testing, an algorithm that would trade based off the AIS alone was able of keeping up with or even outperforming a buy-and-hold investor under certain circumstances. The period used for testing was in general very bullish, therefore the likelihood of a trade's outcome being positive was fairly high.

To fully assess the effectiveness of the AIS, we must also analyze its performance during bearish phases of market. To inspect this further, we expanded the timeframe until June 30, 2022. Between December 31, 2021, and June 30, 2022, Dogecoin had been in a rather strict downwards trend, losing about 60% of its value, going from a price of roughly 0.17 dollars down to 0.069 dollars. We ran the algorithm again, this time with the extended duration, and experimented with different BTFs again to verify our hypothesis that a lower threshold performs better on average.

BTF	End of Day	1 Day	2 Days	3 Days	1 Week	2 Weeks	Average
2	293600,25	157491,79	198758,22	45562,52	66288,61	101457,78	143859,86
5	228060,76	90148,97	141761,02	37669,09	57413,15	108398,08	110575,18
8	149102,80	72460,12	137300,99	35887,72	97236,86	116256,15	101374,11
10	130040,23	64381,45	129655,15	33694,02	100812,99	118553,71	96189,59
15	94060,30	56841,13	113301,85	27920,42	91614,37	122537,04	84379,19
20	81634,41	45627,62	102171,23	25875,32	87061,43	113010,06	75896,68
30	53730,80	36804,37	94674,67	26570,71	94392,93	113208,24	69896,95
50	31277,04	24695,52	65302,05	22006,53	70824,96	111696,57	54300,45
100	18632,55	16510,32	41049,19	17450,89	41207,52	103923,79	39795,71

Table 9: Results of the trading algorithm trading Dogecoin; started on Jan 5 2021, balances recorded on Jun 30, 2022

As you can see in Table 9, the lower BTFs still performed noticeably better, confirming our hypothesis that lower BTFs perform better in general. We reason that a lower BTF is superior even in bearish conditions because the price tends to rise in the short term after an influential tweet, regardless of general market trend. The end-of-day seller was able to capture some of these small moves to the upside, while maintaining no exposure to the asset while its price was generally falling.

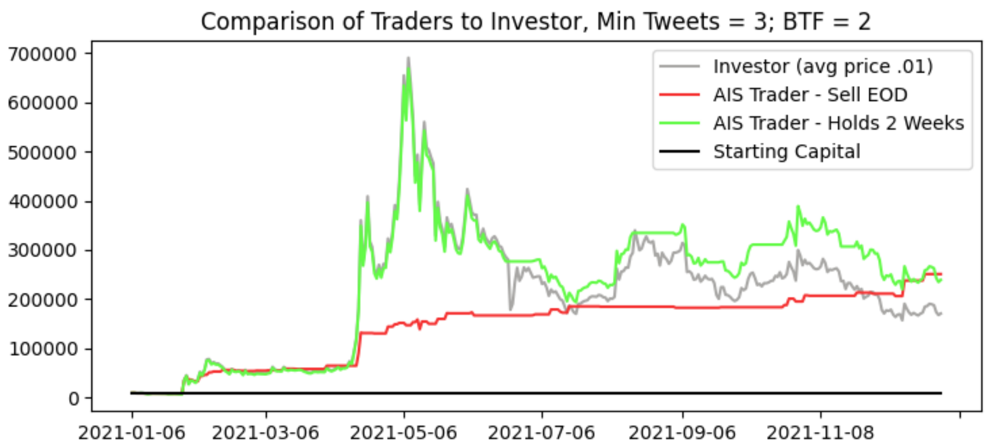

Figure 21: The results of the trading algorithm on December 31, 2021, trading Dogecoin; compared to an investor that bought at the same time as the trading algorithm started, Minimum Tweets = 3; BTF = 2

Figure 21 shows, how our optimal configuration (Window Size = 48, Minimum Tweets = 3, BTF = 2) had performed over the analyzed period from January 05, 2021, until December 31, 2021. As you can see, both AIS traders were able to outperform the buy-and-hold investor. As mentioned, this period could overall be considered overall very bullish. The results become much more interesting when the timeframe is extended by 6 months of general market downtrend.

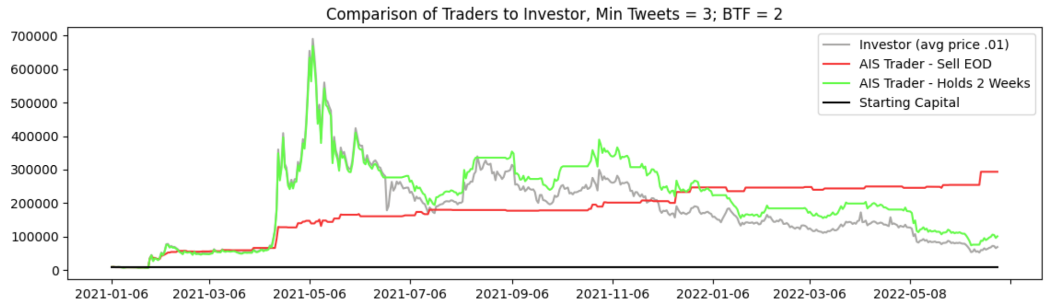

Figure 22: The results of the trading algorithm on June 30, 2022, trading Dogecoin; compared to an investor that bought at the same time as the trading algorithm started, Minimum Tweets = 3; BTF = 2

Figure 22 shows that the trading algorithm was able to not only retain but increase its balance when selling at the end of each trading day, even during times of a general downward trend. This shows that the AIS can be a powerful and useful trading indicator for assets which's price is correlated to twitter activity.

The results differ when the asset is less or hardly related to a tastemakers twitter activity. With the same parameters, the trading algorithm was barely able to achieve any capital gains trading Bitcoin (3,8%), with the trader that holds for 2 weeks losing roughly 18% of the balance, which is shown in Figure 23.

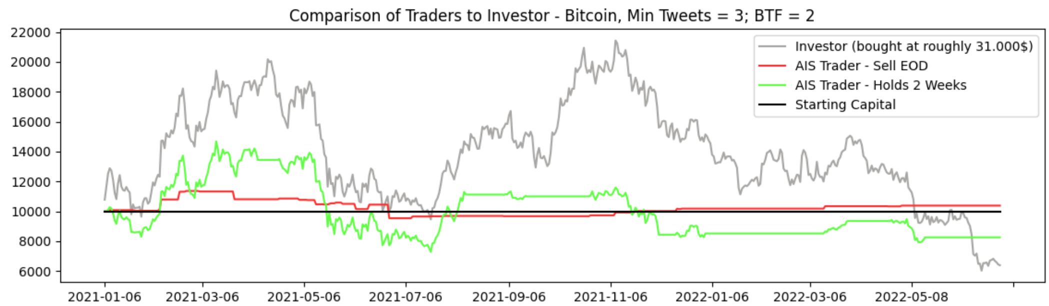

Figure 23: The results of the trading algorithm on June 30, 2022, trading Bitcoin; compared to an investor that bought at the same time as the trading algorithm started, Minimum Tweets = 3; BTF = 2

Our trading algorithm still managed to outperform the buy-and-hold investor at the end regardless, but this was not the case throughout most of the trading period and only happened because Bitcoins price went below the starting price, while the algorithms portfolio consisted of stable assets.

What we do want to highlight is that in both cases (Dogecoin and the much less Twitter correlated Bitcoin), the simulated trader that solely relied on the AIS as a short-term buying indicator was able to steadily increase or maintain their balance, while the investor experienced high volatility. The AIS trader also had very little opportunity cost associated with their investment, as their capital was held liquid during most of the trading period. We therefore deem the AIS to be a valid and powerful indicator when buying and selling purely upon tweets issued by influential users, especially for assets that suggest strong influence from Twitter.

For future research we propose experimenting with multiple assets at the same time. As the end-of-day-seller experienced phases without trading, this capital could've been deployed to multiple assets during the same trading period to potentially further increase the trader's balance.

We also would've wanted to repeat the experiment with Shiba Inu (SHIB), another dog-themed meme-cryptocurrency heavily influenced by Elon Musk. The problem was gathering reliable price data, as Shiba Inu only became tradable on reputable exchanges very late into its popularity. Gathering reliable hourly price data suitable for an academic analysis unfortunately proved to be non-feasible.

Lastly, we want to go over the application which performs the data fetching, dataframe creation and the calculation of the AIS.

6.7 The Client Application

We have applied the principles of Design Science in Information Research by Hevner et al. [11] to create a client application to calculate and evaluate the AIS. The client performs all necessary tasks involved in computing the AIS, like data fetching, data preparation and analysis and will serve as a reference implementation for computing the AIS.

We have created a simple GUI client with which the user can interact. The client was written in Java 17, we used Amazon's Corretto 17 JDK and openjfx 17.0.1. The client code is fully open source and can be found at https://git01lab.cs.univie.ac.at/university_research/masterarbeiten/masterarbeit-krypto-behaviour-prediction-kevin-miller.

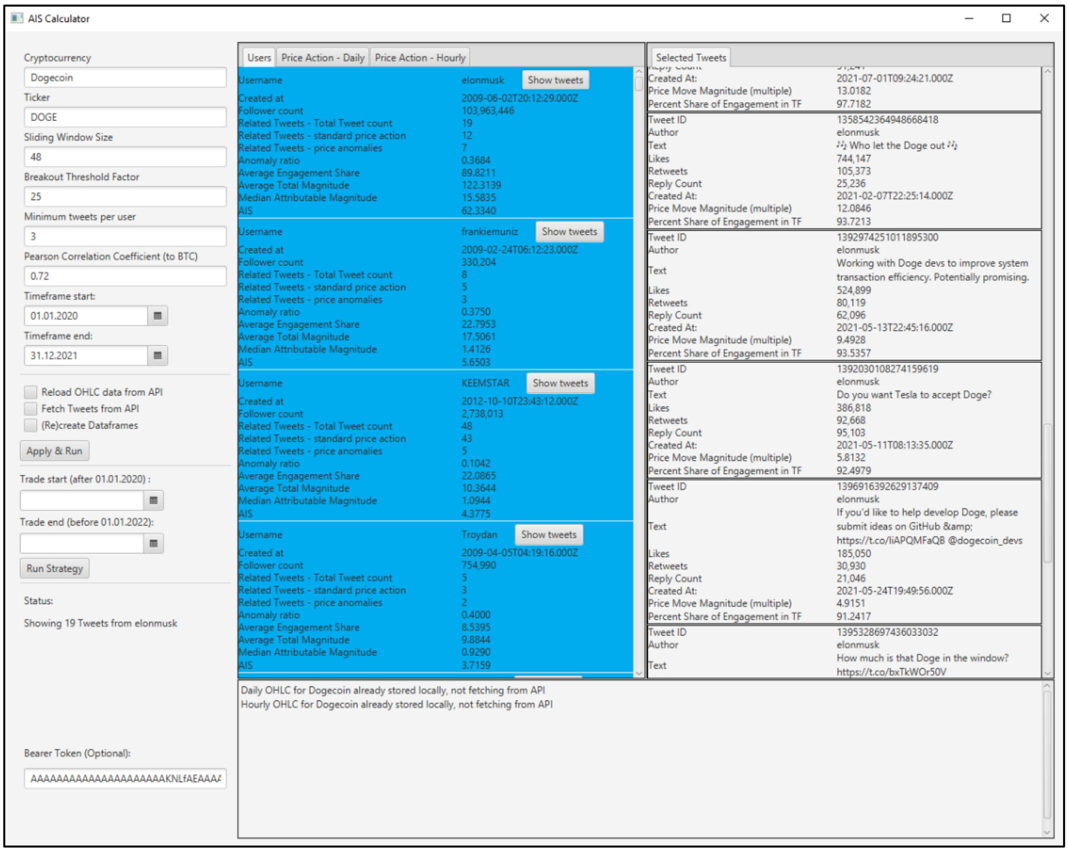

Figure 24: The client application after computing the AIS scores for December 31, 2021, showing Tweets from Elon Musk

Figure 24 shows a screenshot of the entire client application displaying the AIS scores and some selected tweets. The application is structured into four main areas. On the left the control panel is located, where the user can input and manipulate parameters which will then be used to perform calculations and fetch data. Next to the control panel is the overview, which displays information about users as well as price action. Next to the overview is the panel for viewing the selected tweets. The displayed tweets are selected through the overview. On the bottom is the console which displays from the application and informs the user about events or state changes.

6.7.1 Control Panel

The control panel acts as the input area for the application. Here, the user can customize the parameters for computing the AIS. A screenshot of the control panel can be seen in Figure 25.

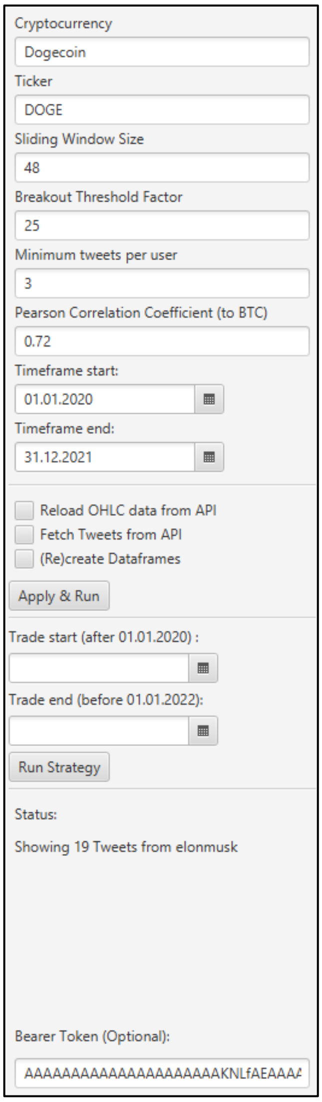

Figure 25: The control panel of the client application

Cryptocurrency
The name of the cryptocurrency that shall be analyzed. *default = Dogecoin*

Ticker
The ticker / trading symbol of the cryptocurrency.
default = DOGE

Sliding Window Size
The size of the sliding window.
default = 48 (proven to be most effective)

Breakout Threshold Factor
The breakout threshold factor used in the AIS computation. *default = 8*

Minimum Tweets per User
The minimum number of tweets a user must have issued before being considered for AIS computation. We only use tweets from officially verified users.
default = 3

Pearson Correlation Coefficient (to BTC)
The PCC between the cryptocurrency and bitcoin, default = 0.72, which is the PCC of dogecoin to bitcoin at the time of writing. The PCC is used to adapt the attributable magnitude, which is essential for calculating the AIS. The user must look up the PCC of the desired cryptocurrency manually[19], as we did not find a way to automatically retrieve this information.

Timeframe start:
The start of the timeframe for which data is fetched and dataframes are created. *Default = 01.01.2020*

Timeframe end:
The date for which the AIS's should be computed. Must lie within the available time range for OHLC & tweet data. *Default = 31.12.2021*

[19] https://cryptowat.ch/de/correlations [accessed October 2022]

Reload OHLC data from API
Deletes any locally stored OHLC data and fetches it again from the Cryptowatch API.

Fetch Tweets from API
Checks if tweets are stored locally, starting the *timeframe start* and ending at the *timeframe end*. If an hourly period is missing, it fetches the data from the Twitter API. This requires the Bearer Token field to contain a valid bearer token only obtainable by applying for academic Twitter API access!

(Re)create Dataframes
Deletes and recreates the locally stored dataframes. This option should be ticked if the sliding window size changes, as the statistics stored within the dataframes are calculated for a specific window size.

Apply & Run
Applies the current settings and starts calculating the AIS for all verified users in the given timeframe.

Trade start
Used to set the start date from which trading the AIS will be simulated.

Trade end
The date until the AIS trading simulation will run.

Run Strategy
Starts simulating the AIS trading strategy.

Status
A status message that relays the most important application events (more abstract than the console).

Bearer Token
Here the user can input their bearer token which they got from Twitter to fetch tweets from the Twitter API.

6.7.2 Overview

The overview is split into three tabs and displays either user information or price action information.

User information

This field displays the most relevant information created for each user, based on the `TweetMap` created for the user and can be seen in Figure 26. The meaning of the individual fields have already been explained in the previous section. *Show tweets* displays all the users tweet in the selected tweets section which will be demonstrated shortly.

Username	elonmusk	Show tweets
Created at	2009-06-02T20:12:29.000Z	
Follower count	103,963,446	
Related Tweets - Total Tweet count	19	
Related Tweets - standard price action	12	
Related Tweets - price anomalies	7	
Anomaly ratio	0.3684	
Average Engagement Share	89.8211	
Average Total Magnitude	122.3139	
Median Attributable Magnitude	15.5835	
AIS	62.3340	

Figure 26: Example of a field containing user information in the client application

Price information

This field displays metrices and calculations for a specific timeframe. The orange background color means that this timeframe has been identified as an anomaly, otherwise the background would be white. It contains the OHLCV data for each timeframe as well as the metrices that are used for calculating the AIS of users, most importantly the MEF at the bottom. The user can also press *show tweets* to display all tweets from the respective timeframe. An example can be seen in Figure 27.

Timeframe	2021-04-01 10:00 - 11:00	
Open	0.0540	
High	0.0667	
Low	0.0539	
Close	0.0630	
Volume	202,080,066.1800	
Price Move in Percent	16.5525	
Velocity (High vs. Low * Volume)	259.2687	
Mean Volume for current Window	2,909,246.4404	
Mean Velocity for Window	0.2562	
Current Anomaly Threshold	6.4039	
Current Influenceability Score	357.8004	
Magnitude (in Multiple)	1,012.1457	
Magnitude of Bitcoin	0.8337	
Mag attr. to external Factors	1,011.5454	Show tweets

Figure 27: Example of a price action field (anomaly) in the client application

6.7.3 Selected Tweets

The Selected Tweets displays all tweets for the selected information field, either all tweets from a specific user or all tweets from a specific timeframe. Figure 28 shows an example of this. When displayed for a user, they are sorted by magnitude, when displayed for a timeframe, they are sorted by engagement share.

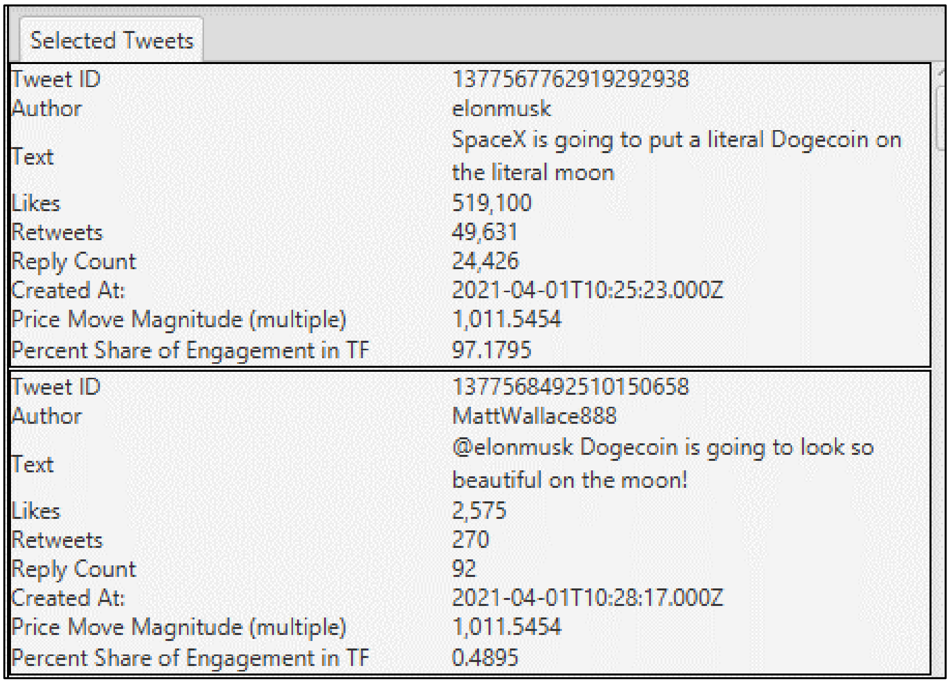

Figure 28: the client application displaying all tweets for in the timeframe 2021-04-10 10:00 - 11:00

6.7.4 Console

The console outputs updates about the current state of the application to the user, for example when data is fetched.

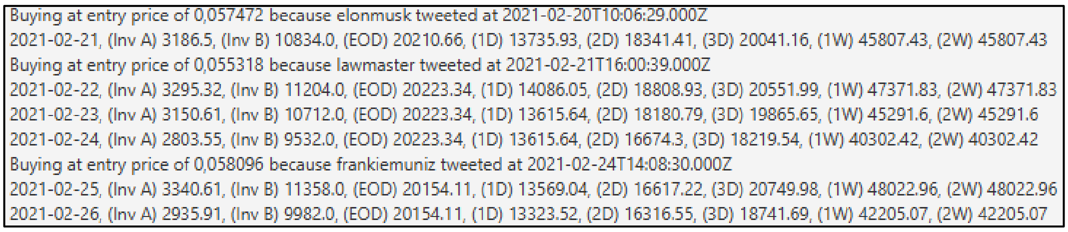

Figure 29: The trading algorithm running in the client

In Figure 29 you can see a snippet of the trading algorithm being executed in the client. The client informs the user about buys executed and tracks the current balances for all investors/ traders. The trading algorithm is discussed in the next section.

6.7.5 Data

The client uses the already described dataframes to perform all calculations, which must be created initially. The creation process takes place automatically.

An important note is that the repository version of the client does not contain the data necessary to compute the AIS for any cryptocurrency. The reason is that it is against Twitter's terms of service to publicly store Twitter data, especially since the historical data we use is not available to the public but instead requires foregoing a special application process to gain access to. We have decided to not take the risk of having legal repercussions by making the data available publicly. It is however possible for anybody to apply for academic access and use the bearer-token provided by Twitter to download tweets. When provided with a valid bearer-token, the client will perform all necessary data fetching automatically.

7 Conclusion

We have performed a Systematic Literature Review based on the work of Barbara Kitchenham [12] to assess the current state of the art in the field of cryptocurrency prediction using microblogging data. We had posed 4 literature related research questions (RQ), which we were able to answer.

First, we analyzed current approaches and algorithms for price prediction which utilize data from microblogs and tried to identify any areas that lack research (RQ 1). We determined that current approaches almost exclusively rely on sentiment analysis, overall tweet volume or search volume, while sourcing the data from a rather broad userbase. The authors used neural networks and autoregressive prediction models, which were effective but rather difficult to understand and hardly replicable.

For RQ 2 we analyzed the extent to which price predictions use sentiment analysis and found that sentiment analysis played a negligible role when it comes to prediction accuracy. Many authors found that sentiment analysis alone was not a viable indicator at all. The best results were achieved when incorporating multiple streams of data, like search and tweet volume.

Next, we looked at how data is typically fetched, prepared, processed and presented, which was analyzed for RQ 3. The Twitter API was used for fetching tweets very similarly to the approach documented in our research. Some authors further performed standardization techniques on the text, like removing punctuation, stop words, URLs and lemmatizing the words. We omitted this step because we did not analyze tweet content. Presenting such large volume of textual data in graphical form is difficult, but many authors added figures and illustrations showcasing e.g. prediction accuracy.

The last research question (RQ 4) which was answered through the literature research was whether the results are meaningful, explainable and replicable. While many authors do make significant claims of being able to achieve above market returns, they fail to provide full transparency. Many papers simply do not document how their algorithm makes the decisions it makes and fail to provide any instructions on how their research could be replicated.

This also led us to our final research question (RQ 5), which focuses on the explainability and replicability aspects of this field of research. We asked whether it was possible to create meaningful metric for classifying the potential influence of a given user over a given asset in a transparent manner. To answer this question, we developed the Asset Influence Score or AIS, an effective yet explainable, easily understandable and fully replicable metric for assessing the suggested influence a Twitter user has over the price of an asset. The AIS was created and documented in a way which allows every researcher to replicate and expand upon our interpretation of a simple yet effective method for microblogging-based trading and influence assessment. We do this by providing detailed documentation about every step involved in the process, as well as providing a full open-source reference implementation of a client capable of computing the AIS. We have also proven with experiments on Dogecoin that it would've been possible to outperform a buy-and-hold investor in the long-term using only the AIS as a trading indicator.

Due to its rather simplistic nature, the metric does come with its share of shortcomings. In the current version of the AIS, we assume that twitter activity represents the sole force behind the "external factor" part of our calculations, which is most certainly not the case. For future work, one must certainly thoroughly assess the actual influence that twitter as a platform has over the markets.

Another shortcoming, not of the AIS but of the data, is that the Twitter API does not allow deleted tweets to be retrieved. We only have access to the tweets that the users deliberately chose to keep public, which limits the thoroughness of our assessment capabilities. There is unfortunately no way to retrieve deleted tweets in retrospect, at least not officially. To circumvent this, one must permanently fetch and store all recently issued tweets about an asset.

Also, this should go without saying, but the AIS does not perform nearly as well on assets that do not show signs of correlation to Twitter. While still managing to outperform a buy-and-hold investor on Bitcoin, we believe this to be a coincidence rather than an achievement by the AIS. Large assets like Bitcoin simply aren't impacted enough through Twitter to utilize the AIS effectively.

For future work, we would like to identify other assets that are strongly correlated to the actions of a single Twitter user, but these are rather difficult to come by and none, except for the aforementioned Shiba Inu (which we could not analyze due to the lack of reliable price data), come close to being such a good test subject as Dogecoin. Testing for such currencies would require much more capable API access, as we were heavily limited by both speed and the tweet limits imposed by the Academic Twitter API. Running the trading simulations also requires rather powerful hardware, since the algorithm has to iterate over Gigabytes of dataframe-data for each day. This could most likely be optimized or reimplemented in a more efficient programming language than Java.

We would also like to experiment with Sentiment Analysis. Even though literature research suggests that SA is not that powerful as a price prediction indicator, it might be useful for the AIS as it is heavily dependent on the social response to influential users.

Lastly, network effects play a vital role in the study of information diffusion and influence analysis. Identifying the root of influence is important, as this identification can potentially forecast tweets by more influential Twitter users from which they drew inspiration from. If an identified tastemaker draws heavy inspiration from a lesser known and arguably less influential source, one could argue that the original content was responsible for causing the market move, if it had to be reposted either directly or indirectly.

As a first step for identifying network effects, we propose identifying relationships based on retweets and replies. Strong connections could be identified by detecting frequent retweets and/or replies among the same userbase.

Further work could be incorporate a solution based on Latent Dirichlet Allocation for topic detection, timestamps and the relationship (following / not following) between users.

Algorithm:

- Perform LDA with large number of topics on every tweet within a timeframe using the sliding window approach
- If a distinct topic can be identified, find the first tweet within the sliding window that mentions that topic
- If a tweet from a certain topic causes a significant market move, we credit the original author as being partially responsible for the market move.

Abstract

English – Bitcoin went from being a niche topic among cypherpunks to a global investment vehicle within only a decade, establishing itself firmly in the sector as a solid financial instrument. Then, when billionaire and Tesla CEO Elon Musk joked on Twitter about a fun dog-themed cryptocurrency, the world looks in awe as prices shot through the roof. Dogecoin saw an astonishing 15.000% gain within only a couple of months of being Musk's favorite topic to tweet about. Dogecoin was started and intended as a joke with no real value, yet it had reached a valuation of multiple billions of dollars. The price of Dogecoin was inevitably tied to Musk's Twitter account.
In this paper we propose the Asset Influence Score (AIS), a metric to quantify the suggested influence a Twitter user has over the price of a cryptocurrency. The AIS is a score between 0 and 100 and describes the certainty of with which a newly issued tweet by a user will co-occur in a period of elevated trading activity.
The AIS will arm investors with the knowledge about which persons tweets are seemingly valued highly by the market, allowing for efficient monitoring of trading positions. We also evaluate the AIS as a valuable trading indicator, showing that it would've been possible to outperform a buy-and-hold Dogecoin investor if one had traded based on solely the AIS as a buy-signal, while also maintaining higher liquidity compared to a buy-and-hold investor.
Index Terms – Bitcoin, Dogecoin, Cryptocurrency, Price prediction, Twitter, Influence analysis

Deutsch – Bitcoin hat sich innerhalb von nur einem Jahrzehnt von einem Nischenthema unter Cypherpunks zu einem globalen Anlageinstrument entwickelt und sich in der Branche als solides Finanzinstrument etabliert. Als dann der Milliardär und Tesla-CEO Elon Musk auf Twitter einen Witz über eine Hunde-Kryptowährung machte schossen die Preise nach oben. Dogecoin verzeichnete einen erstaunlichen Anstieg von 15.000 % innerhalb von nur ein paar Monaten, während Musk regelmäßig darüber twitterte. Dogecoin wurde als Scherz ohne wirklichen Wert ins Leben gerufen und hatte dennoch eine Bewertung von mehreren Milliarden Dollar erreicht. Der Kurs von Dogecoin war unweigerlich mit Musks Twitter-Account verknüpft.
In diesem Werk schlagen wir den Asset Influence Score (AIS) vor, eine Metrik zur Quantifizierung des vermuteten Einflusses eines Twitter-Nutzers auf den Preis einer Kryptowährung. Der AIS ist ein Wert zwischen 0 und 100 und beschreibt die Sicherheit, mit der ein neu veröffentlichter Tweet eines Nutzers in einer Periode erhöhter Handelsaktivität auftritt.
Der AIS gibt Anlegern das Wissen darüber, welche Tweets von Personen vom Markt stark gewichtet werden, und ermöglicht so eine effiziente Überwachung von Handelspositionen. Wir bewerten den AIS auch als Trading-Indikator und zeigen, dass es möglich gewesen wäre, einen "Buy-and-Hold"-Dogecoin-Investor an Rendite übertreffen, wenn man nur auf der Grundlage des AIS als Kaufsignal gehandelt hätte.
Schlagwörter – Bitcoin, Dogecoin, Kryptowährung, Preisprognose, Twitter, Einflussanalyse

Bibliography

[1] “Live stock, index, futures, Forex and Bitcoin charts on TradingView,” *TradingView*. https://www.tradingview.com/symbols/SPX/ (accessed May 13, 2022).

[2] “Vergleich der Immobilienpreise verschiedener Städte.” https://www.immopreise.at/Preisvergleich (accessed May 13, 2022).

[3] D. / A. Hamacher, “The man who dodged the Dogecoin,” *Decrypt*, Oct. 16, 2018. https://decrypt.co/3722/dogecoin-inventor-jackson-palmer-regrets-nothing (accessed May 15, 2022).

[4] R. K. Aggarwal and G. Wu, “Stock Market Manipulations,” *J. Bus.*, vol. 79, no. 4, pp. 1915–1953, 2006, doi: 10.1086/503652.

[5] N. Gandal, J. T. Hamrick, T. Moore, and T. Oberman, “Price manipulation in the Bitcoin ecosystem,” *J. Monet. Econ.*, vol. 95, pp. 86–96, 2018, doi: 10.1016/j.jmoneco.2017.12.004.

[6] M. Cary, “Down with the #Dogefather: Evidence of a Cryptocurrency Responding in Real Time to a Crypto-Tastemaker,” *J. Theor. Appl. Electron. Commer. Res.*, vol. 16, no. 6, pp. 2230–2240, Sep. 2021, doi: 10.3390/jtaer16060123.

[7] T. Hoang and J. Mothe, “Predicting information diffusion on Twitter - Analysis of predictive features,” *J. Comput. Sci.*, vol. 28, pp. 257–264, Sep. 2018, doi: 10.1016/j.jocs.2017.10.010.

[8] S. Mahdizadehaghdam, H. Wang, H. Krim, L. Dai, and IEEE, “INFORMATION DIFFUSION IN INTERCONNECTED HETEROGENEOUS NETWORKS,” in *University of North Carolina*, 2017, pp. 3759–3763.

[9] F. Victor and T. Hagemann, “Cryptocurrency Pump and Dump Schemes: Quantification and Detection,” in *2019 International Conference on Data Mining Workshops (ICDMW)*, Nov. 2019, pp. 244–251. doi: 10.1109/ICDMW.2019.00045.

[10] C. Vinopal, “Kim Kardashian is being sued for allegedly misleading investors about a crypto scheme,” *Quartz*. https://qz.com/2112251/kim-kardashian-is-being-sued-over-the-ethereummax-cryptocurrency/ (accessed Jun. 05, 2022).

[11] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design Science in Information Systems Research,” p. 32.

[12] B. Kitchenham, “Procedures for Performing Systematic Reviews,” *Keele UK Keele Univ*, vol. 33, Aug. 2004.

[13] M. Ye and G. Li, “Internet big data and capital markets: a literature review,” *Financ. Innov.*, vol. 3, no. 1, Dec. 2017, doi: 10.1186/s40854-017-0056-y.

[14] A. Java, X. Song, T. Finin, and B. Tseng, “Why we twitter: understanding microblogging usage and communities,” in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis - WebKDD/SNA-KDD ’07*, San Jose, California, 2007, pp. 56–65. doi: 10.1145/1348549.1348556.

[15] “Number of Twitter Users 2022/2023: Demographics, Breakdowns & Predictions,” *Financesonline.com*, Mar. 31, 2020.

https://financesonline.com/number-of-twitter-users/ (accessed Jun. 05, 2022).
[16] A. Cavacini, “What is the best database for computer science journal articles?,” *Scientometrics*, vol. 102, no. 3, pp. 2059–2071, Mar. 2015, doi: 10.1007/s11192-014-1506-1.
[17] E. F. Fama, “Efficient Capital Markets: A Review of Theory and Empirical Work,” *J. Finance*, vol. 25, no. 2, pp. 383–417, 1970, doi: 10.2307/2325486.
[18] J. R. Nofsinger, “Social Mood and Financial Economics,” *J. Behav. Finance*, vol. 6, no. 3, pp. 144–160, Sep. 2005, doi: 10.1207/s15427579jpfm0603_4.
[19] S. Yang, S. Mo, and A. Liu, “Twitter financial community sentiment and its predictive relationship to stock market movement,” *Quant. FINANCE*, vol. 15, no. 10, pp. 1637–1656, Oct. 2015, doi: 10.1080/14697688.2015.1071078.
[20] D. Kahneman and A. Tversky, “Prospect Theory: An Analysis of Decision under Risk,” *Econometrica*, vol. 47, no. 2, pp. 263–291, 1979, doi: 10.2307/1914185.
[21] J. Abraham, D. Higdon, J. Nelson, and J. Ibarra, “Cryptocurrency Price Prediction Using Tweet Volumes and Sentiment Analysis,” vol. 1, no. 3, p. 22, 2018.
[22] J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” *J. Comput. Sci.*, vol. 2, no. 1, pp. 1–8, Mar. 2011, doi: 10.1016/j.jocs.2010.12.007.
[23] A. Mittal and A. Goel, “Stock Prediction Using Twitter Sentiment Analysis,” p. 5.
[24] C. Hutto and E. Gilbert, “VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text,” *Proc. Int. AAAI Conf. Web Soc. Media*, vol. 8, no. 1, Art. no. 1, May 2014.
[25] J. Beck *et al.*, “Sensing Social Media Signals for Cryptocurrency News,” in *ETH Zurich*, 2019, pp. 1051–1054. doi: 10.1145/3308560.3316706.
[26] X. Guo and J. Li, “A Novel Twitter Sentiment Analysis Model with Baseline Correlation for Financial Market Prediction with Improved Efficiency,” *2019 Sixth Int. Conf. Soc. Netw. Anal. Manag. Secur. SNAMS*, pp. 472–477, Oct. 2019, doi: 10.1109/SNAMS.2019.8931720.
[27] C. Bhuvaneshwari and R. Beena, “TFCMA-DRL: Tolerant Flexible Coordinated Multi-Agent Deep Reinforcement Learning for Prediction of Future Stock Price Trends from Multi-Source Data,” *Int. J. Intell. Eng. Syst.*, vol. 14, no. 2, pp. 33–42, 2021, doi: 10.22266/ijies2021.0430.04.
[28] S. Mohapatra, N. Ahmed, and P. Alencar, “KryptoOracle: A Real-Time Cryptocurrency Price Prediction Platform Using Twitter Sentiments,” in *University of Waterloo*, 2019, pp. 5544–5551.
[29] G. A. A. J. Alkubaisi, S. S. Kamaruddin, and H. Husni, “Conceptual framework for stock market classification model using sentiment analysis on twitter based on Hybrid Naïve Bayes Classifiers,” *Int. J. Eng. Technol.*, vol. 7, no. 2, pp. 57–61, 2018, doi: 10.14419/ijet.v7i2.14.11156.
[30] Z. Ben-Ami, R. Feldman, and B. Rosenfeld, “Using Multi-View Learning to Improve Detection of Investor Sentiments on Twitter,” *Comput. Sist.*, vol. 18, no. 3, pp. 477–490, Jul. 2014, doi: 10.13053/CyS-18-3-2019.
[31] L. Evans, M. Owda, K. Crockett, and A. Vilas, “A methodology for the resolution of cashtag collisions on Twitter - A natural language processing &

data fusion approach," *EXPERT Syst. Appl.*, vol. 127, pp. 353–369, Aug. 2019, doi: 10.1016/j.eswa.2019.03.019.
[32] A. Fernandez Vilas, R. P. Diaz Redondo, and A. Lorenzo Garcia, "The Irruption of Cryptocurrencies into Twitter Cashtags: A Classifying Solution," *IEEE Access*, vol. 8, pp. 32698–32713, 2020, doi: 10.1109/ACCESS.2020.2973735.
[33] H. Brans and B. Scholtens, "Under his thumb the effect of president Donald Trump's Twitter messages on the US stock market," *PLOS ONE*, vol. 15, no. 3, Mar. 2020, doi: 10.1371/journal.pone.0229931.
[34] P. Gjerstad, P. Meyn, P. Molnar, and T. Naess, "Do President Trump's tweets affect financial markets?," *Decis. SUPPORT Syst.*, vol. 147, Aug. 2021, doi: 10.1016/j.dss.2021.113577.
[35] H. Alostad and H. Davulcu, "Directional prediction of stock prices using breaking news on Twitter," *WEB Intell.*, vol. 15, no. 1, pp. 1–17, 2017, doi: 10.3233/WEB-170349.
[36] F. De Arriba-Perez, S. Garcia-Mendez, J. Regueiro-Janeiro, and F. Gonzalez-Castano, "Detection of Financial Opportunities in Micro-Blogging Data With a Stacked Classification System," *IEEE ACCESS*, vol. 8, pp. 215679–215690, 2020, doi: 10.1109/ACCESS.2020.3041084.
[37] S. Hazem, E. Mohamed, and H. Ali, "WHAT TWITTER CAN TELL US ABOUT THE STOCK MARKET," *ANDULI*, no. 18, pp. 219–242, 2019, doi: 10.12795/anduli.2019.i18.10.
[38] L. Nizzoli, S. Tardelli, M. Avvenuti, S. Cresci, M. Tesconi, and E. Ferrara, "Charting the Landscape of Online Cryptocurrency Manipulation," *IEEE ACCESS*, vol. 8, pp. 113230–113245, 2020, doi: 10.1109/ACCESS.2020.3003370.
[39] H. Alostad, H. Davuleu, and IEEE, "Directional Prediction of Stock Prices using Breaking News on Twitter," in *Arizona State University*, 2015, pp. 523–530. doi: 10.1109/WI-IAT.2015.82.