# AI environment and architecture for decentralized crypto finance

Anton Kolonin
akolonin@aigents.com
Facebook: akolonin
Telegram: akolonin

autonio.foundation

SingularityNET

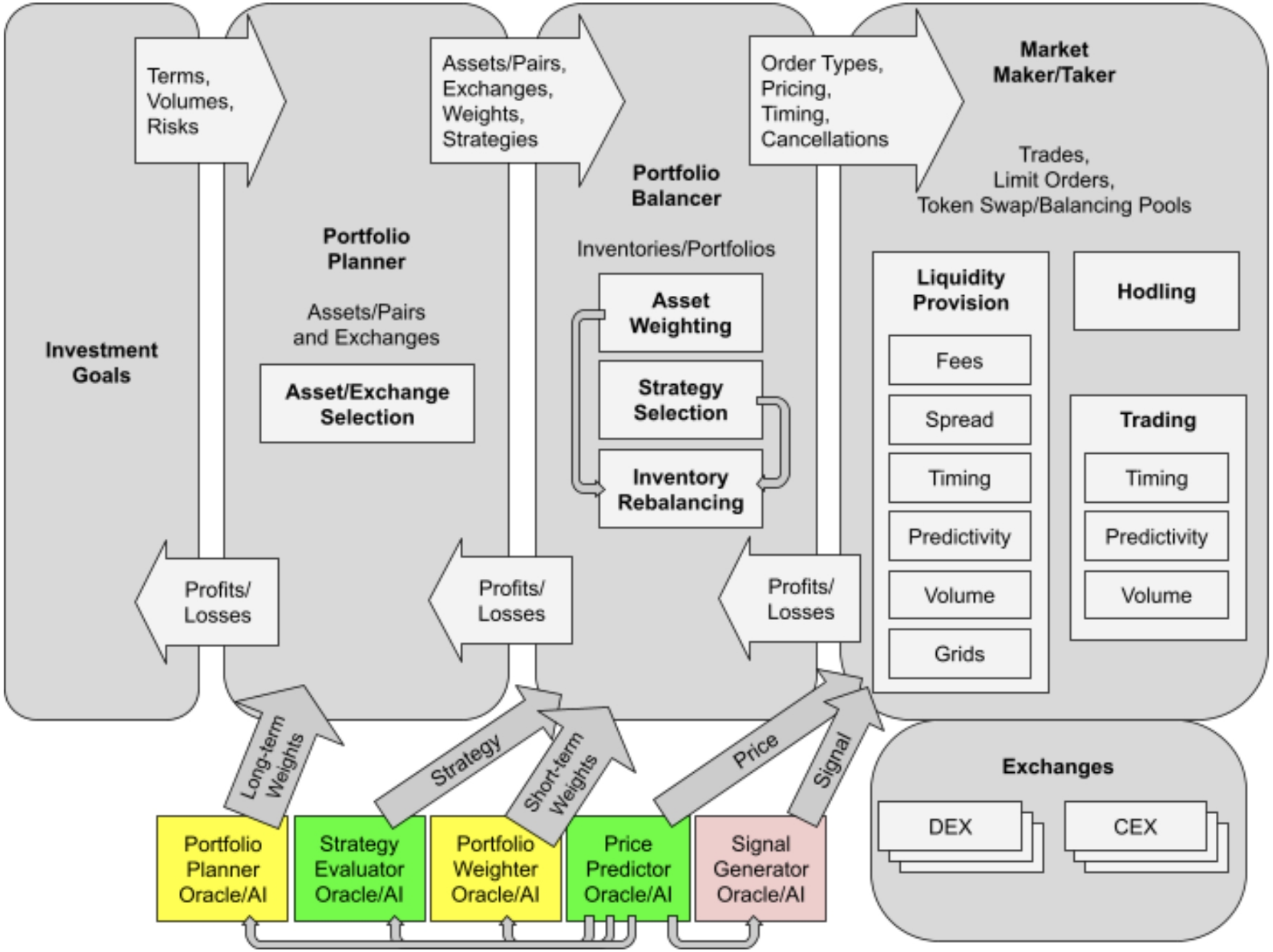
# Optimization Space for Active Portfolio Management

Asset

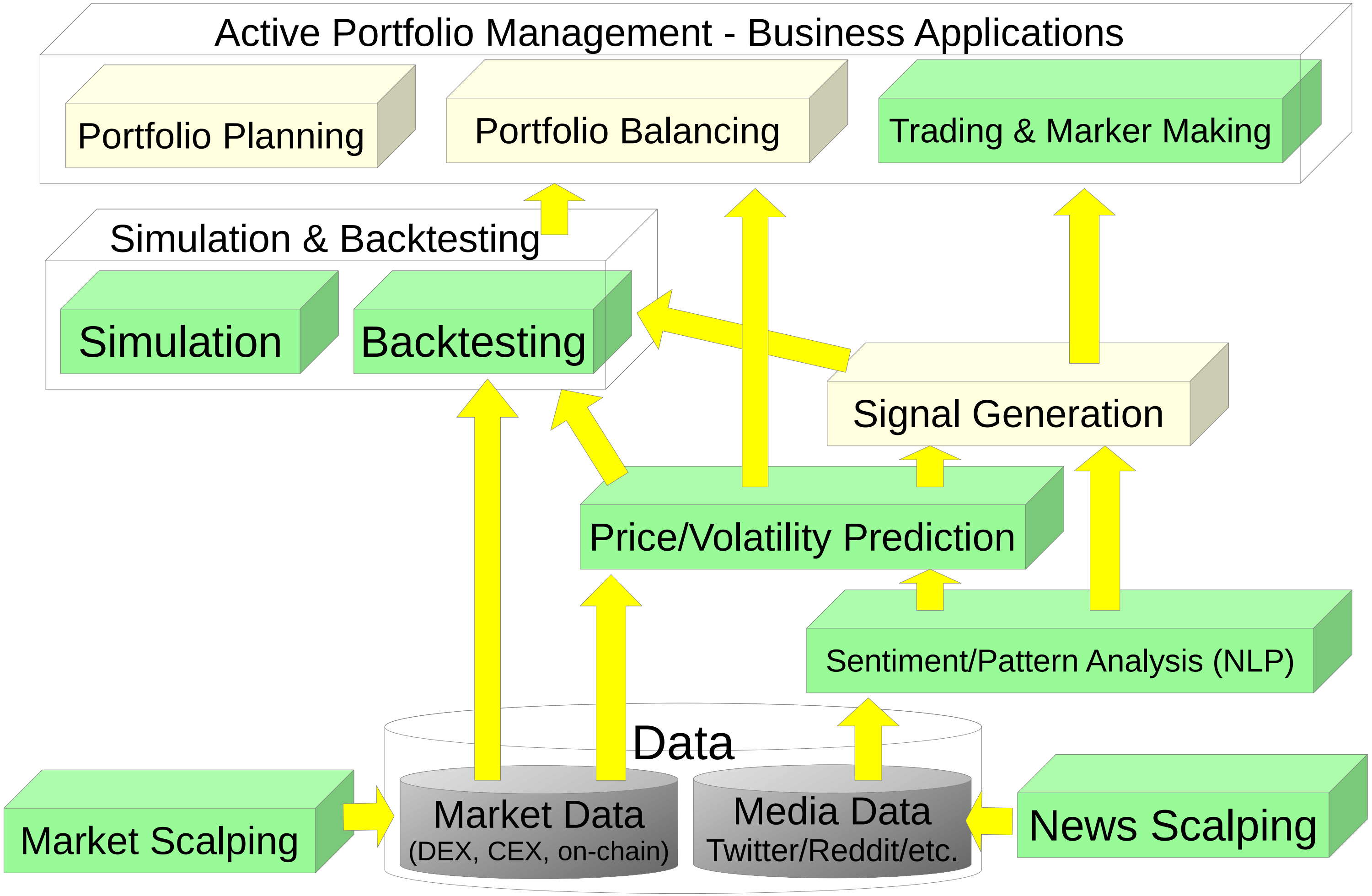
| | **Total** | Hodling | Making Binance | Making Uniswap | Trading Binance | Trading Uniswap |
|---|---|---|---|---|---|---|
| ETH | **30%** | 15% | 5% | | 5% | 5% |
| BTC | **30%** | 5% | 10% | 10% | 5% | |
| AGIX | **20%** | | 10% | 10% | | |
| SDAO | **10%** | | | | 5% | 5% |
| NIOX | **10%** | | 5% | | 5% | |

Macro Strategy

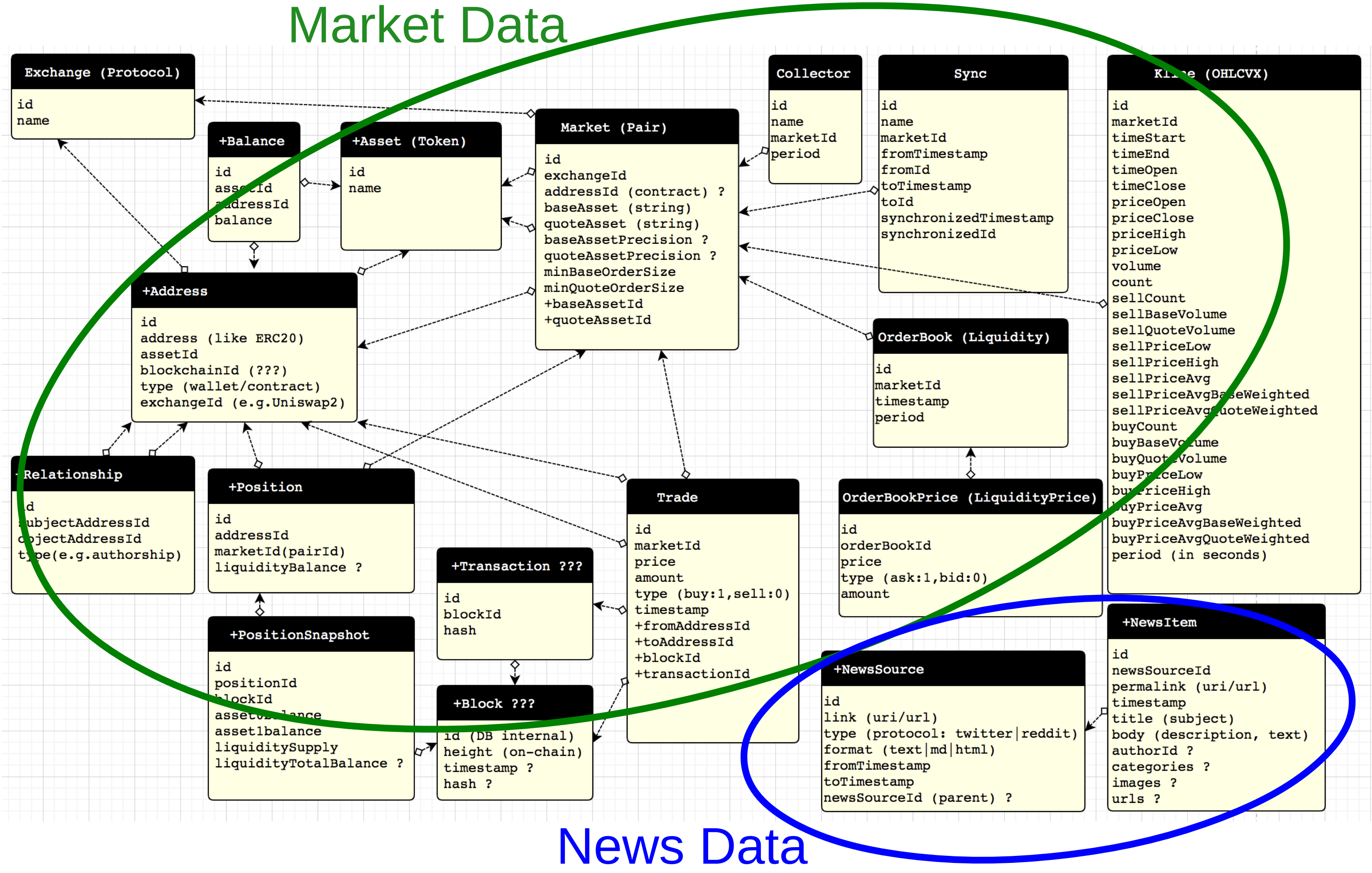
# Active Portfolio Management



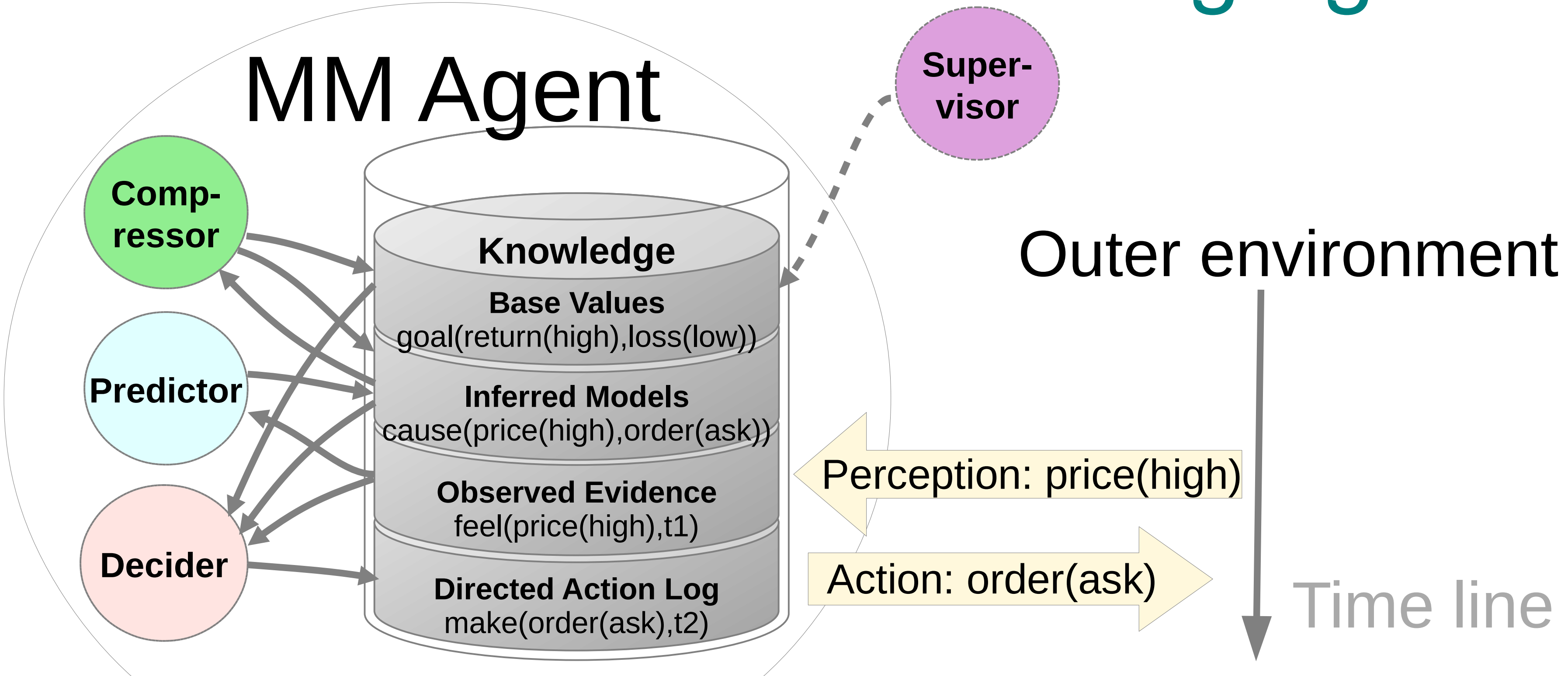
# Services and APIs



# Data


Market Data
Exchange (Protocol)
id
name
+Balance
id
assetId
addressId
balance
+Asset (Token)
id
name
Market (Pair)
id
exchangeId
addressId (contract) ?
baseAsset (string)
quoteAsset (string)
baseAssetPrecision ?
quoteAssetPrecision ?
minBaseOrderSize
minQuoteOrderSize
+baseAssetId
+quoteAssetId
Collector
id
name
marketId
period
Sync
id
name
marketId
fromTimestamp
fromId
toTimestamp
toId
synchronizedTimestamp
synchronizedId
Kline (OHLCVX)
id
marketId
timeStart
timeEnd
timeOpen
timeClose
priceOpen
priceClose
priceHigh
priceLow
volume
count
sellCount
sellBaseVolume
sellQuoteVolume
sellPriceLow
sellPriceHigh
sellPriceAvg
sellPriceAvgBaseWeighted
sellPriceAvgQuoteWeighted
buyCount
buyBaseVolume
buyQuoteVolume
buyPriceLow
buyPriceHigh
buyPriceAvg
buyPriceAvgBaseWeighted
buyPriceAvgQuoteWeighted
period (in seconds)
+Address
id
address (like ERC20)
assetId
blockchainId (???)
type (wallet/contract)
exchangeId (e.g.Uniswap2)
OrderBook (Liquidity)
id
marketId
timestamp
period
+Relationship
id
subjectAddressId
objectAddressId
type(e.g.authorship)
+Position
id
addressId
marketId(pairId)
liquidityBalance ?
Trade
id
marketId
price
amount
type (buy:1,sell:0)
timestamp
+fromAddressId
+toAddressId
+blockId
+transactionId
OrderBookPrice (LiquidityPrice)
id
orderBookId
price
type (ask:1,bid:0)
amount
+Transaction ???
id
blockId
hash
+PositionSnapshot
id
positionId
blockId
asset0balance
asset1balance
liquiditySupply
liquidityTotalBalance ?
+Block ???
id (DB internal)
height (on-chain)
timestamp ?
hash ?
+NewsSource
id
link (uri/url)
type (protocol: twitter|reddit)
format (text|md|html)
fromTimestamp
toTimestamp
newsSourceId (parent) ?
+NewsItem
id
newsSourceId
permalink (uri/url)
timestamp
title (subject)
body (description, text)
authorId ?
categories ?
images ?
urls ?
News Data

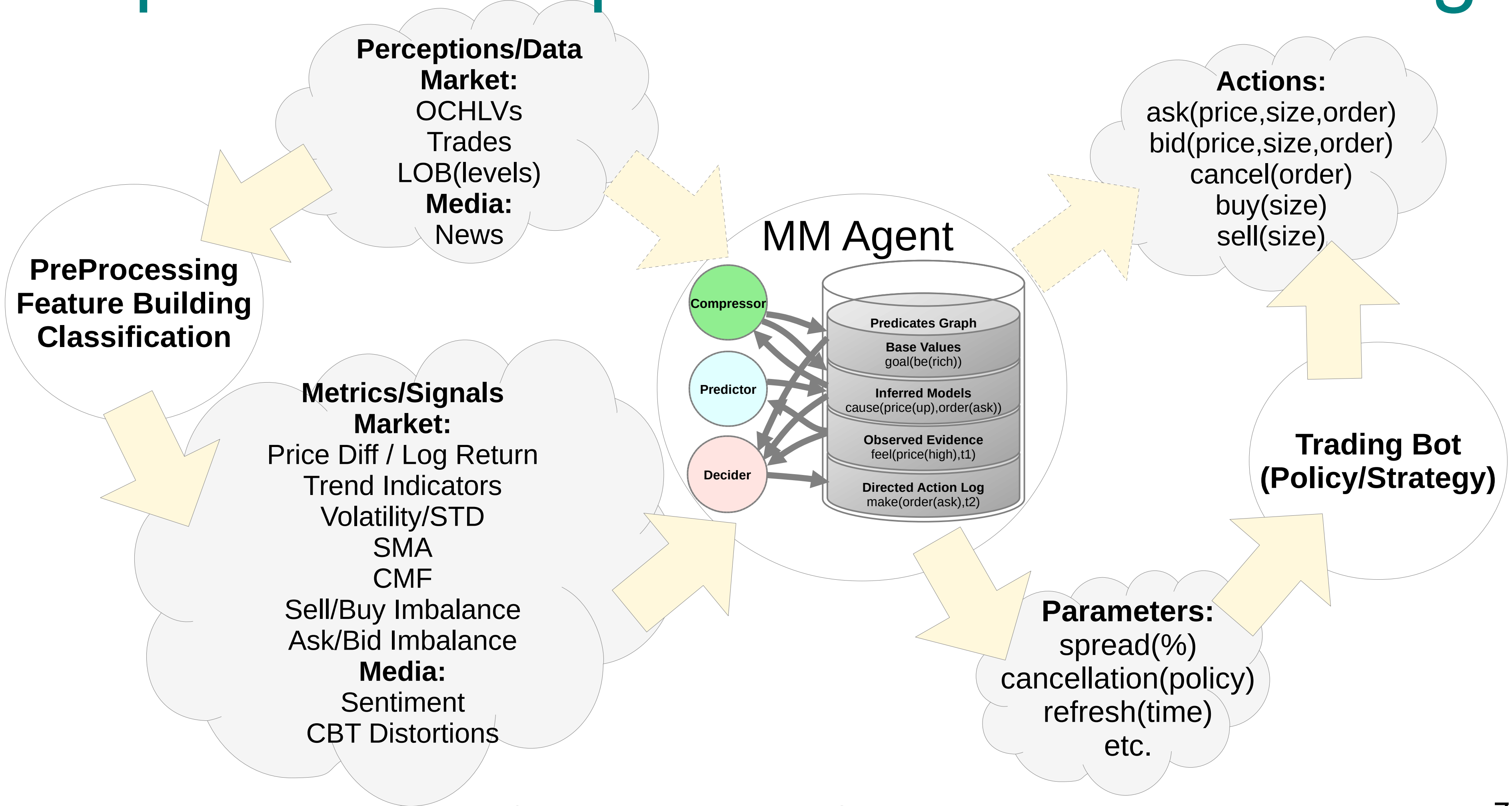
# Narrow AGI for Market Making Agent



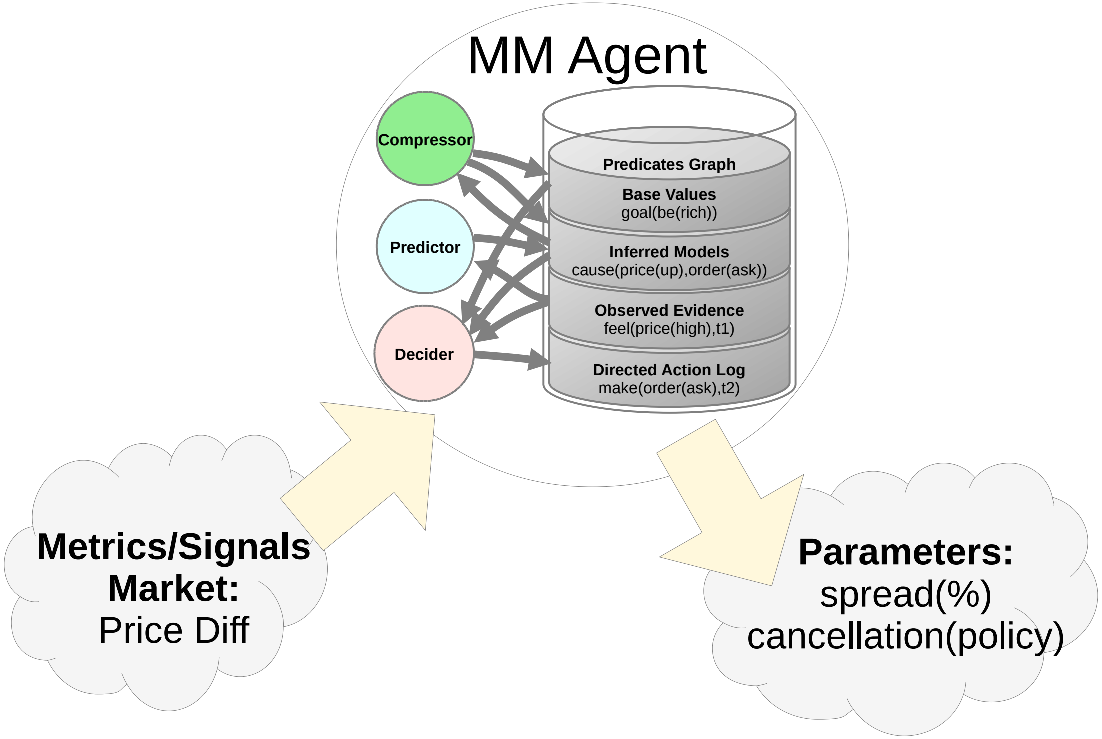

Evgenii E. Vityaev Purposefulness as a Principle of Brain Activity // Anticipation: Learning from the Past, (ed.) M. Nadin. Cognitive Systems Monographs, V.25, Chapter No.: 13. Springer, 2015, pp. 231-254.

Anton Kolonin: Neuro-symbolic architecture for experiential learning in discrete and functional environments // AGI-2021 Conference Proceedings, 2021

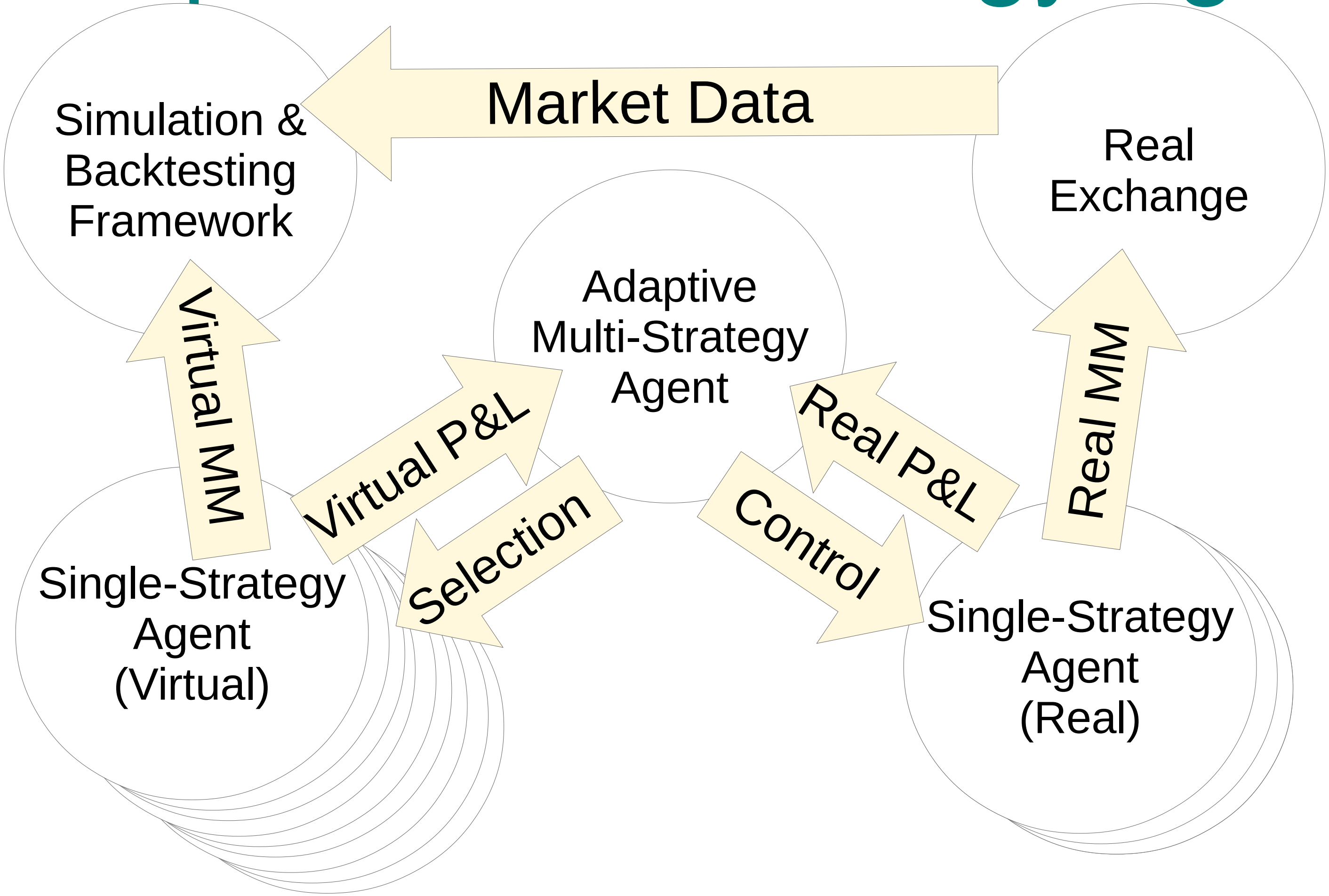
# Operational Space for Market Making



# Operational Space for Market Making (Simplified)

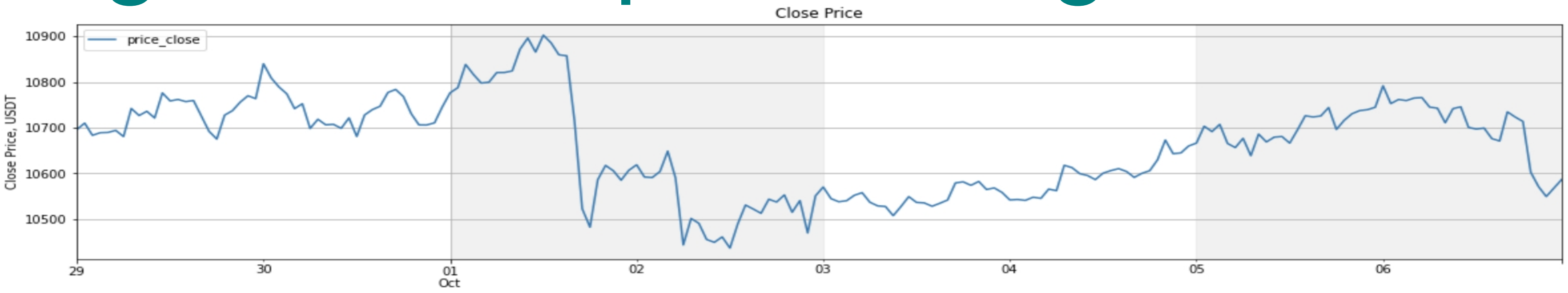


# Adaptive Multi-Strategy Agent

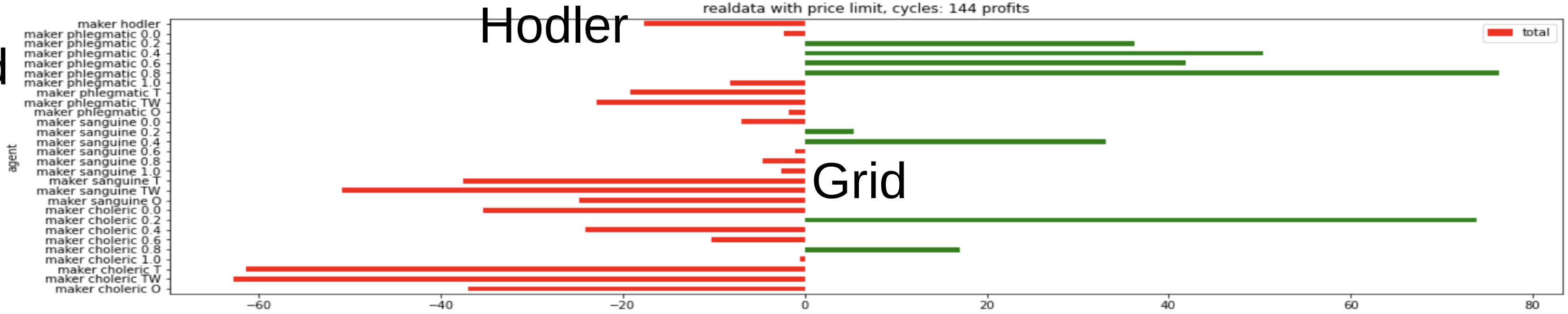


# Trading with multiple Strategies at a Time

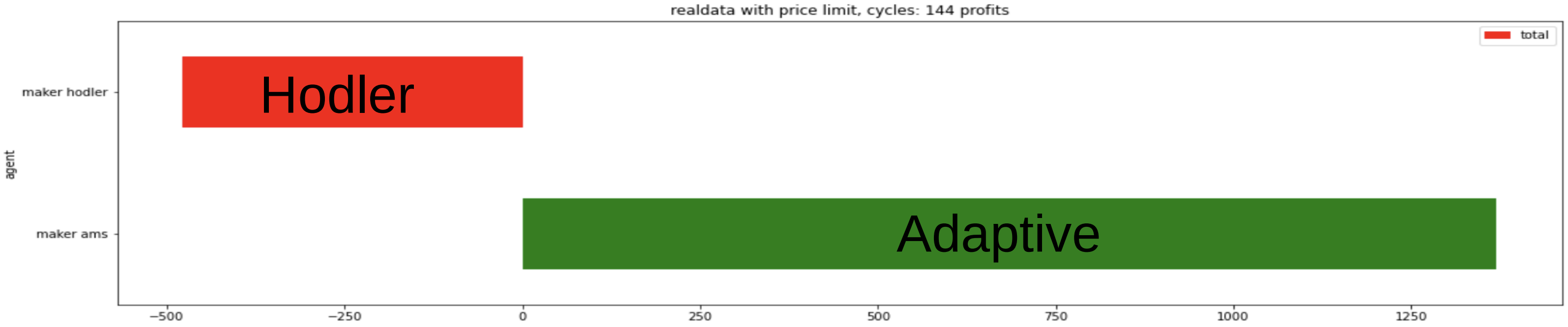
BTC/
USDT




Order Grid
Marker
Making
P & L

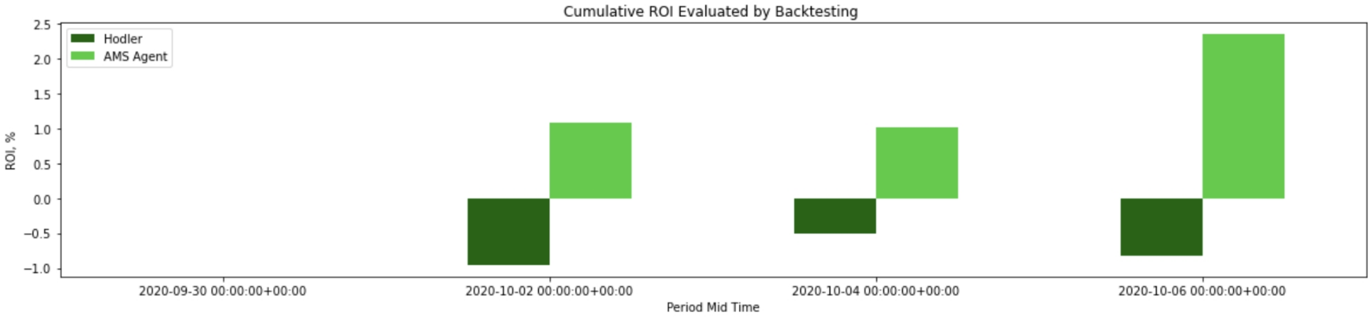



Adaptive
Multi
Strategy
Agent
P & L

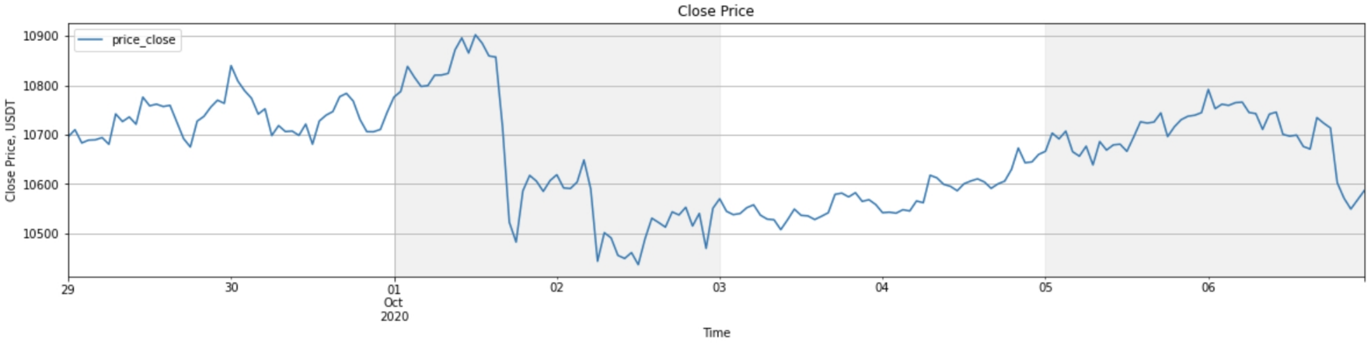


# Incremental ROI for adaptive MM strategy

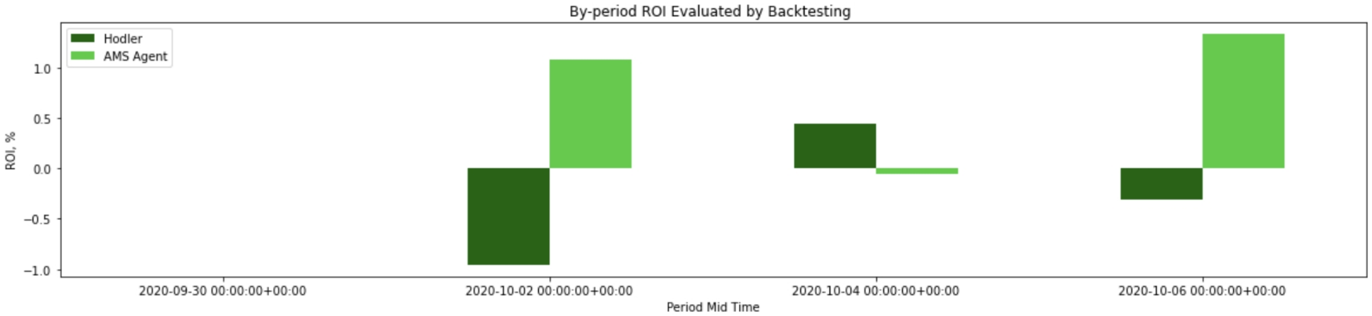
Cumulative
Adaptive
P & L




BTC/
USDT



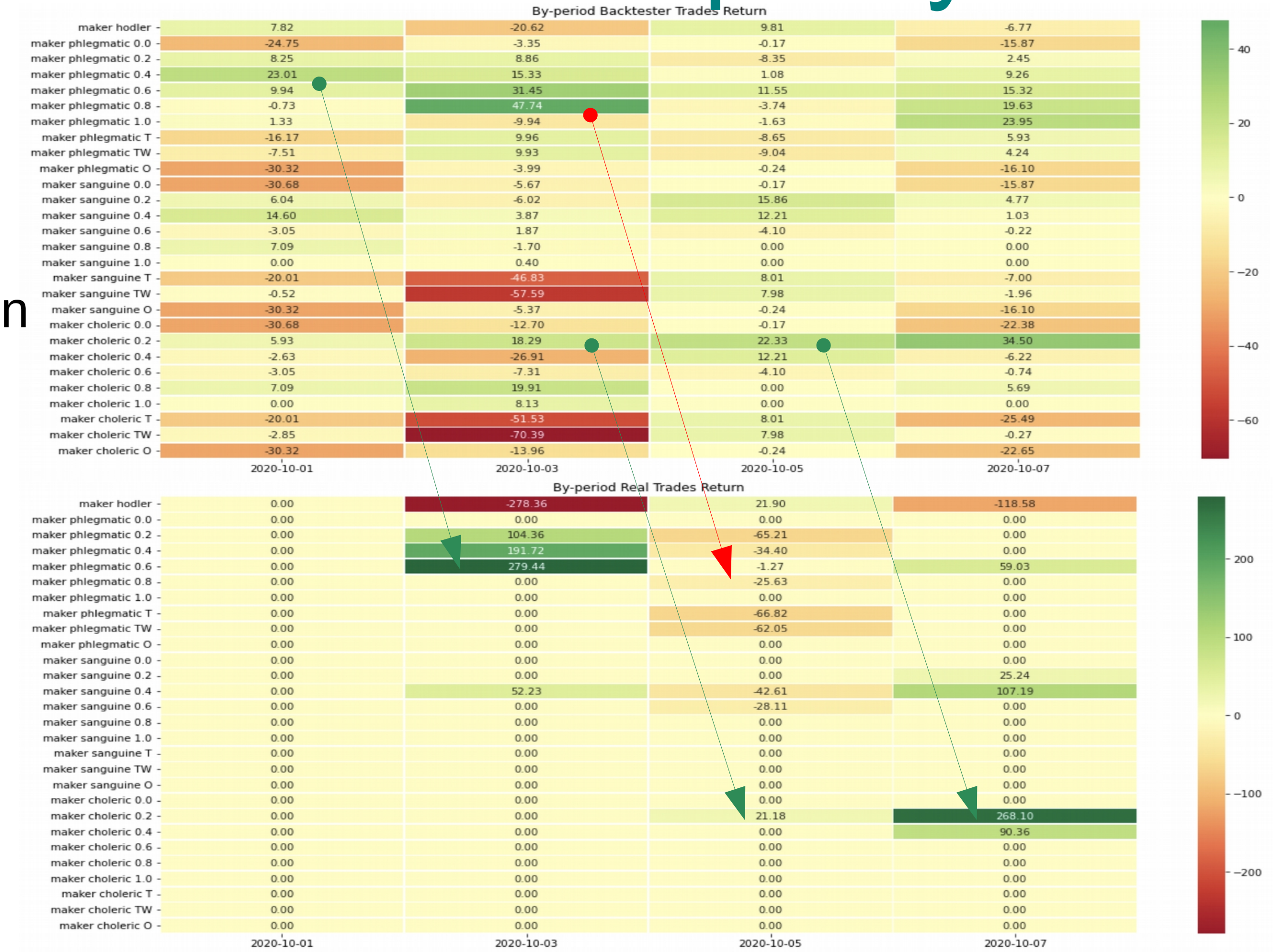

Incremental
Adaptive
P & L



# Real-time model-based policy selection

P & L for Backtesting on historical data / Forward testing on live market data

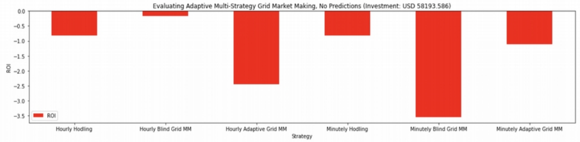
P & L for Trading on live market data



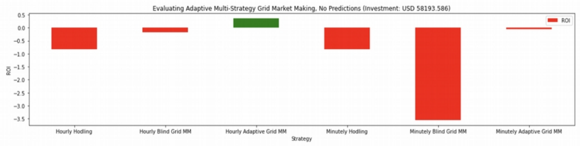
# Overall ROI for adaptive MM strategy

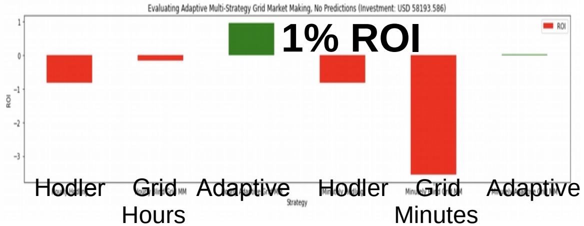
BTC/ USDT


## "Basic/Simple" Strategy



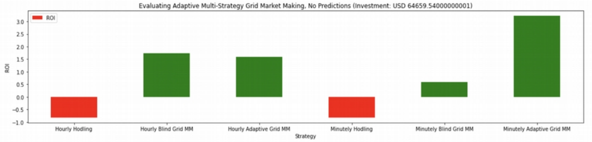

Hodler Grid Hours Adaptive Hodler Grid Minutes Adaptive



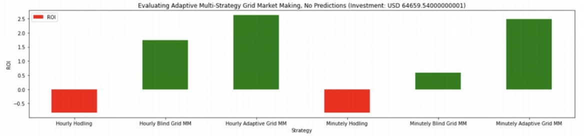




Hodler Grid Hours Adaptive Hodler Grid Minutes Adaptive

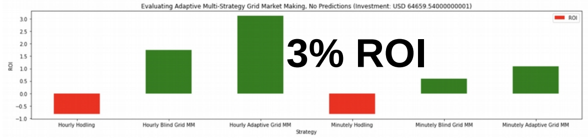
## "NIOX Maker" Strategy




1 day strategy update interval (6 days)

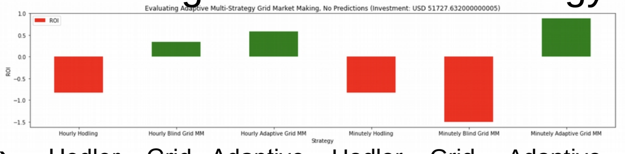



2 days strategy update interval (6 days)

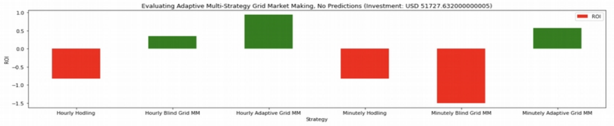


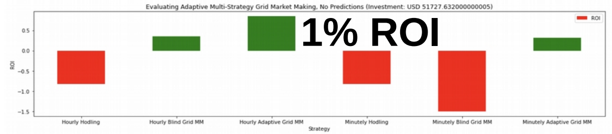

3 days strategy update interval (6 days)

## "Hummingbot Pure MM" Strategy




Hodler Grid Hours Adaptive Hodler Grid Minutes Adaptive

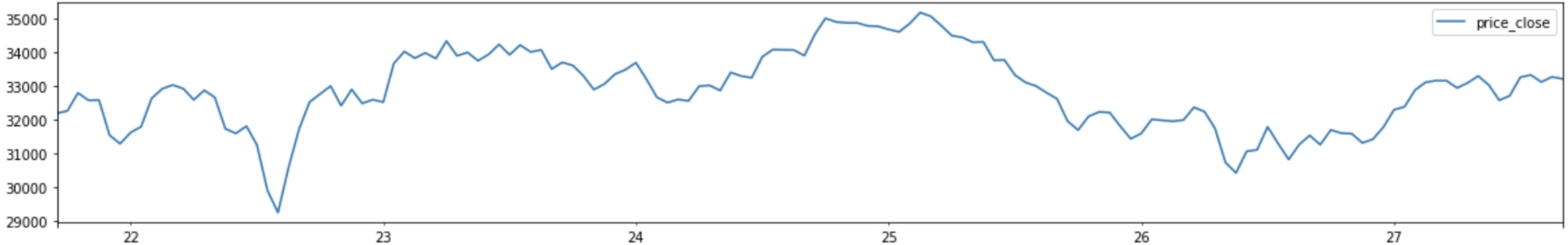


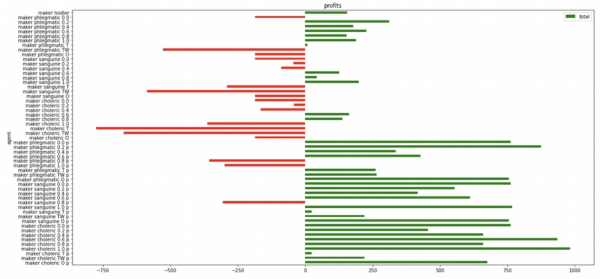

Hodler Grid Hours Adaptive Hodler Grid Minutes Adaptive

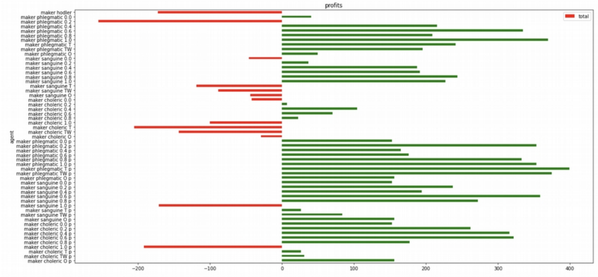
# Non-predictive vs. Predictive (LSTM) MM

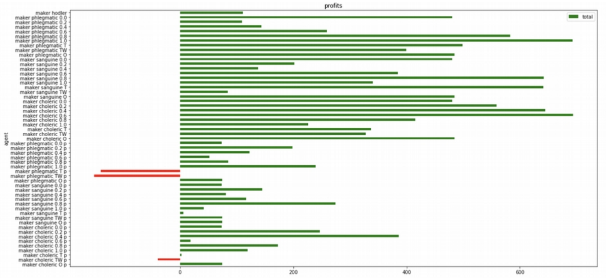
## BTC/USDT June 22-27, 2021



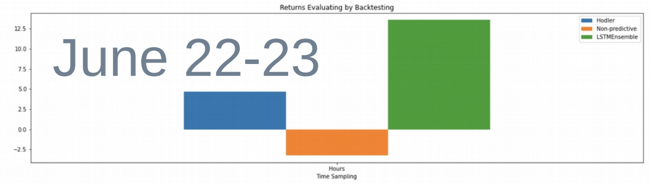

## Profits and Losses (USD) per Market Maker (MM) Agent Strategy

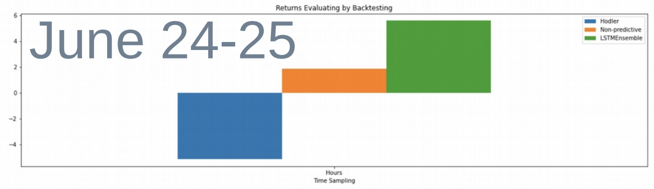


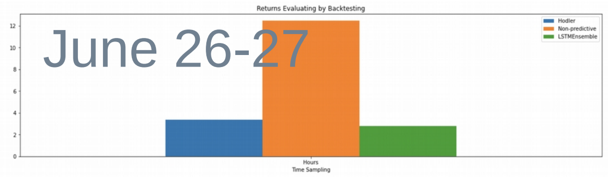




## ROI % per Agent Strategy (Hodler, Non-Predictive MM, Predictive MM)



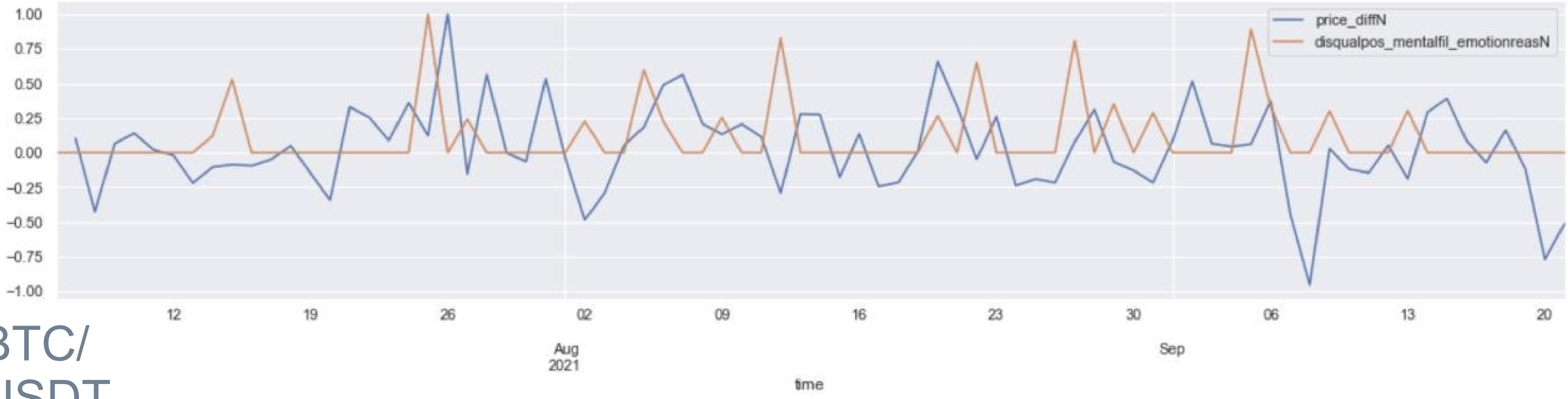



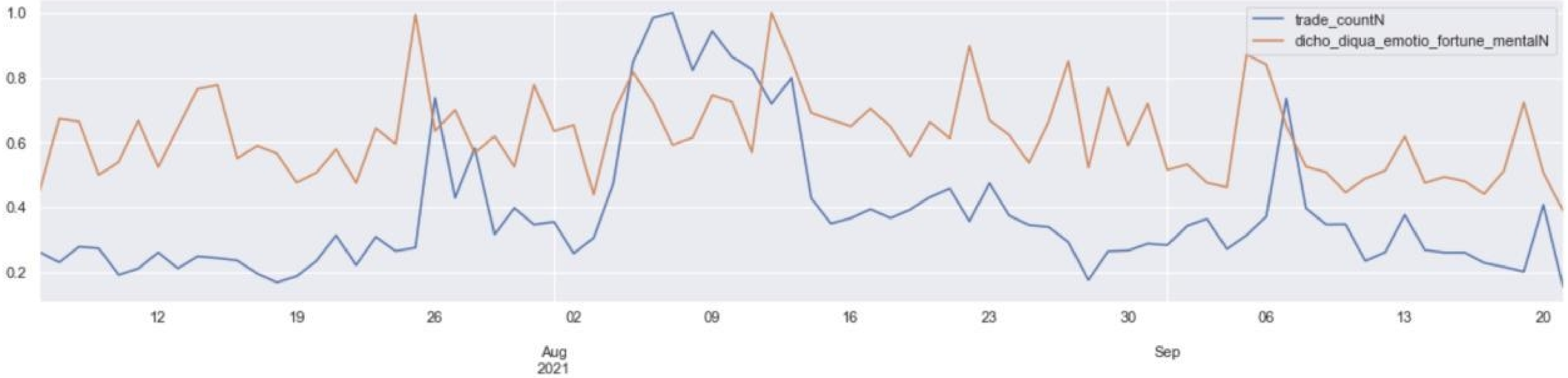
# Sentiment and Behavioral Patterns Mining

On highly-manipulative markets – sentiment and intent and insider information are the best predictors ... if you get them

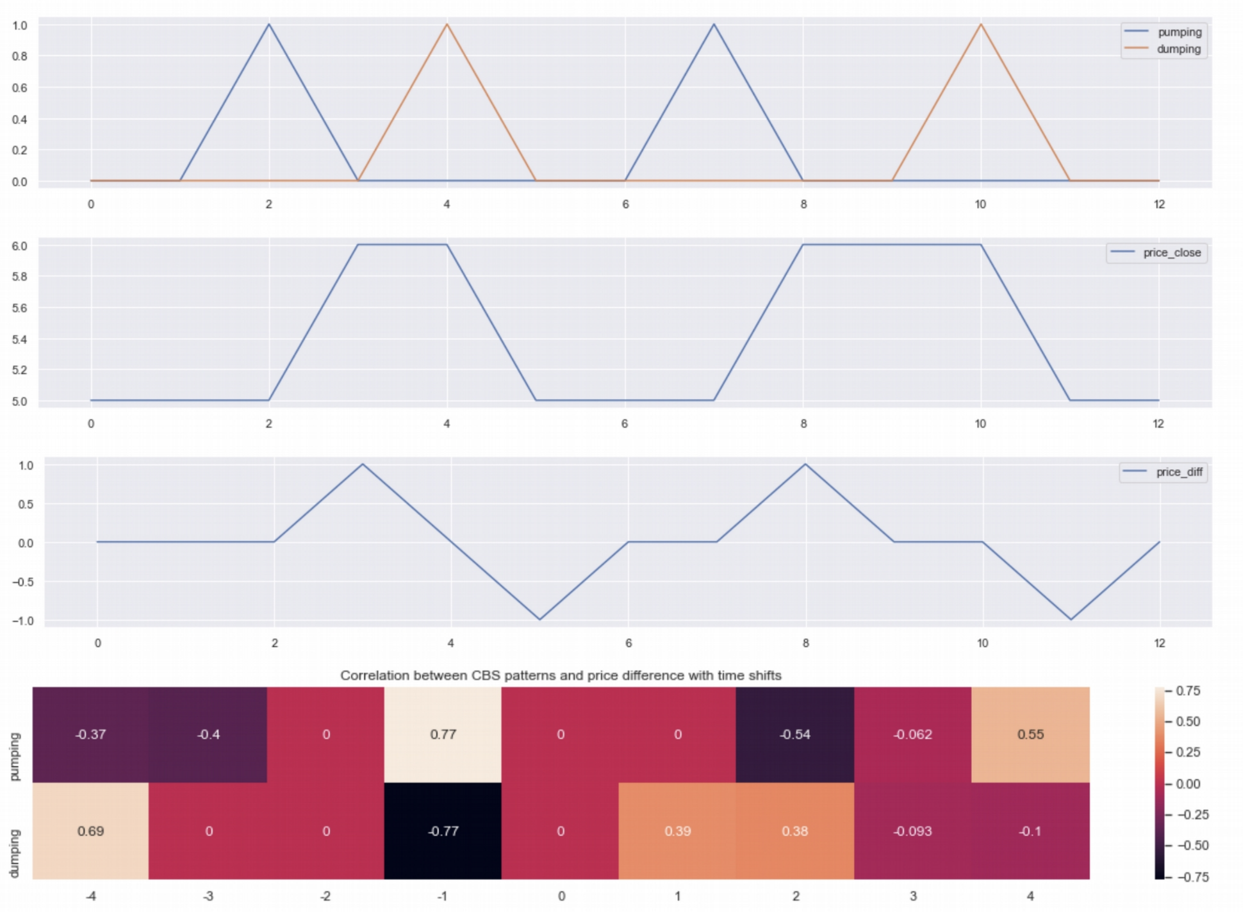

# Emotions & Distortions affecting the Markets




BTC/
USDT

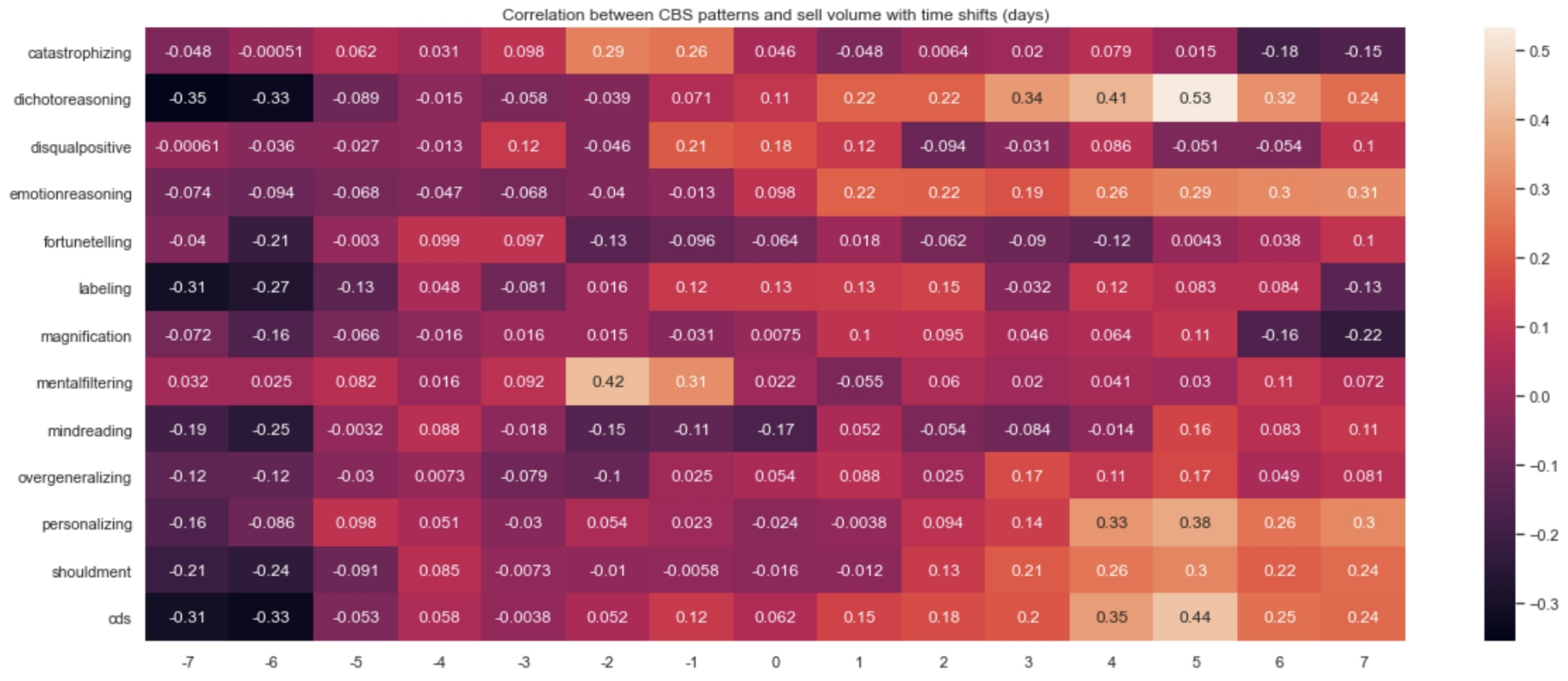


# Causal Indicator Mining for Price Moves




Reverse temporal correlation as causal connection between sentiment patterns and price change

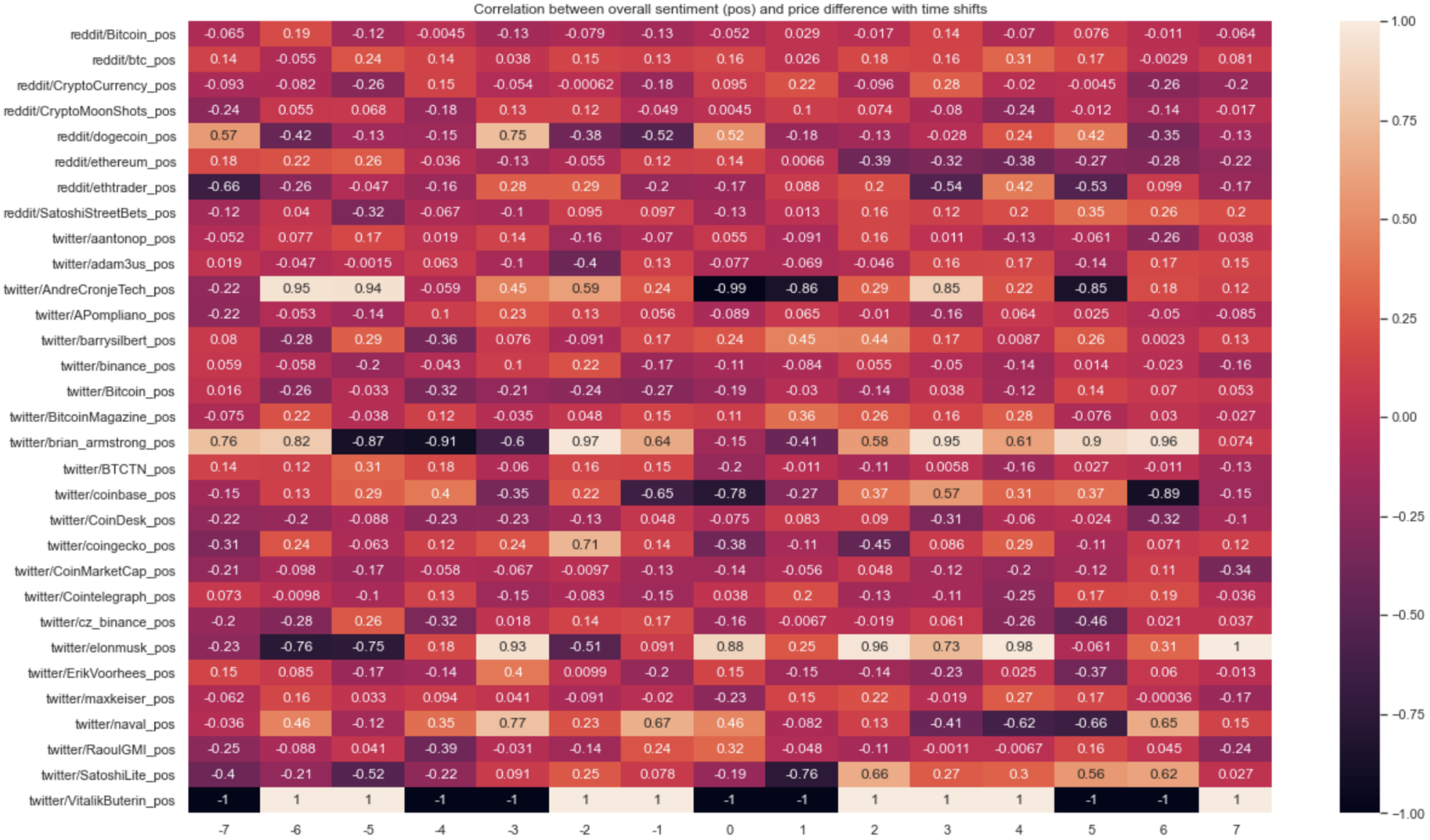
# CBS patterns vs. Market Dynamics (Volume)




Tracking overall Twitter and Reddit cognitive distortions as a predictor for BTC/USD trade volume

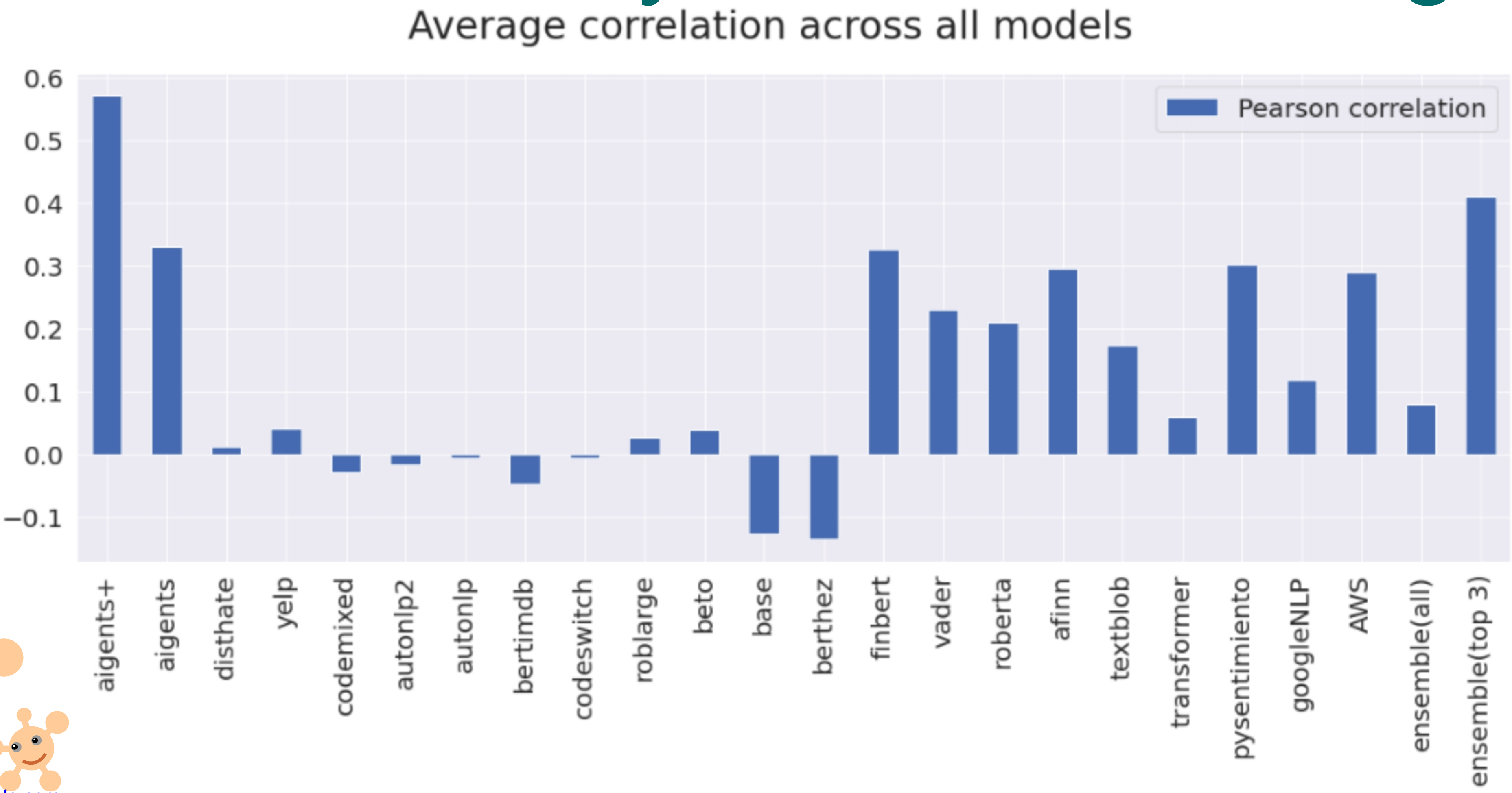
```
#Catastrophizing: Exaggerating the importance of negative events
distortions['catastrophizing'] = "will fail, will go wrong, will end, will be impossible, will not happen...
#Mental Filtering: Paying too much attention to negative details instead of the whole picture
distortions['mentalfiltering'] = "I see only, all I see, all I can see, can only think, nothing good...
```

# Sentiment for Market Price



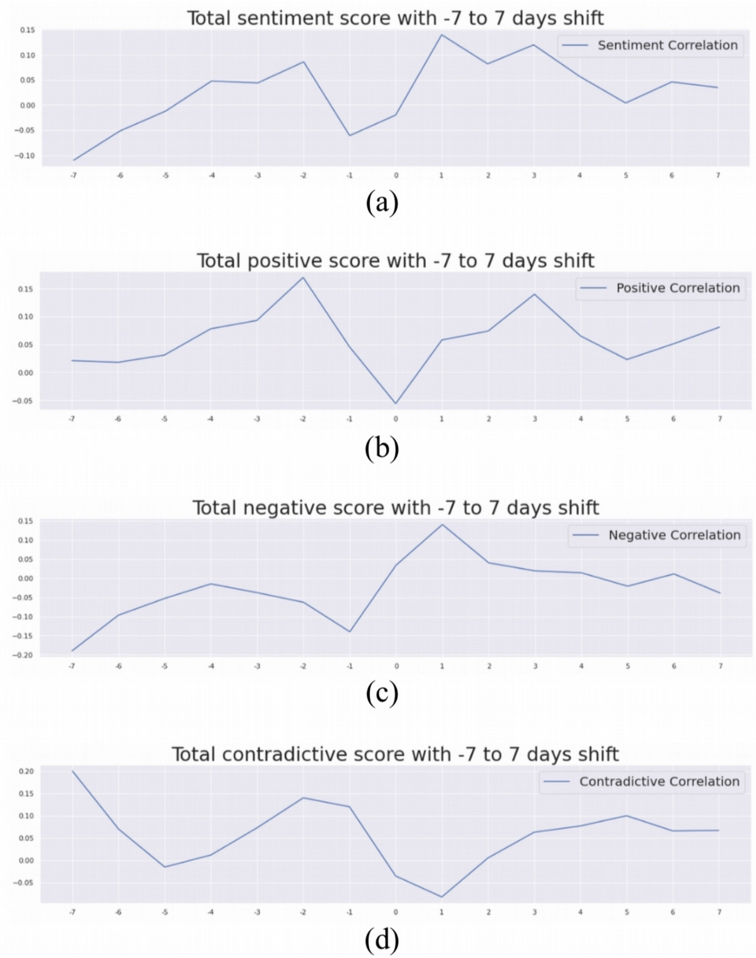

Tracking Twitter and Reddit sentiment on per-channel basis as a predictor for BTC/USD price movements

# Sentiment Analysis – Models' Fight

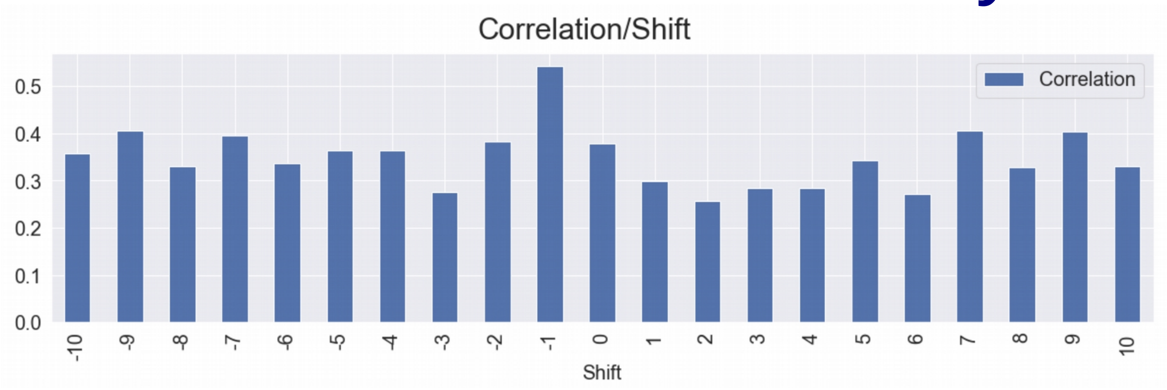



AIGENTS
https://aigents.com

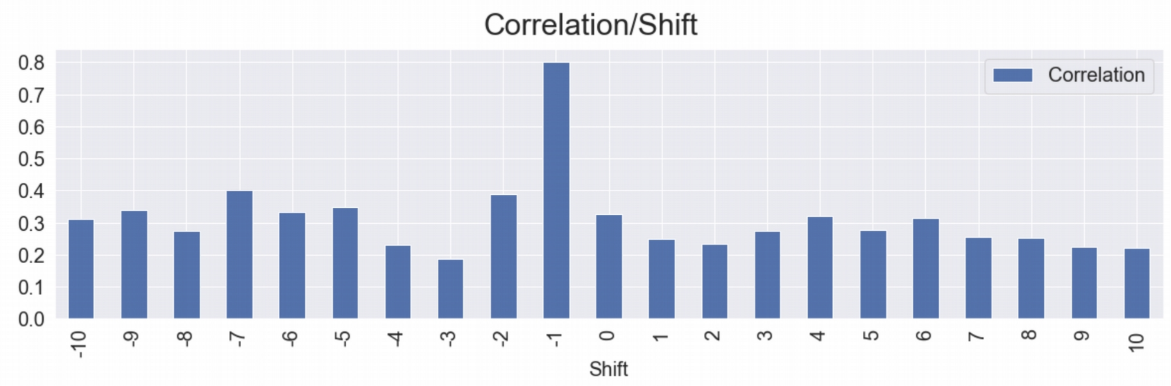
# Temporal Correlational Analysis



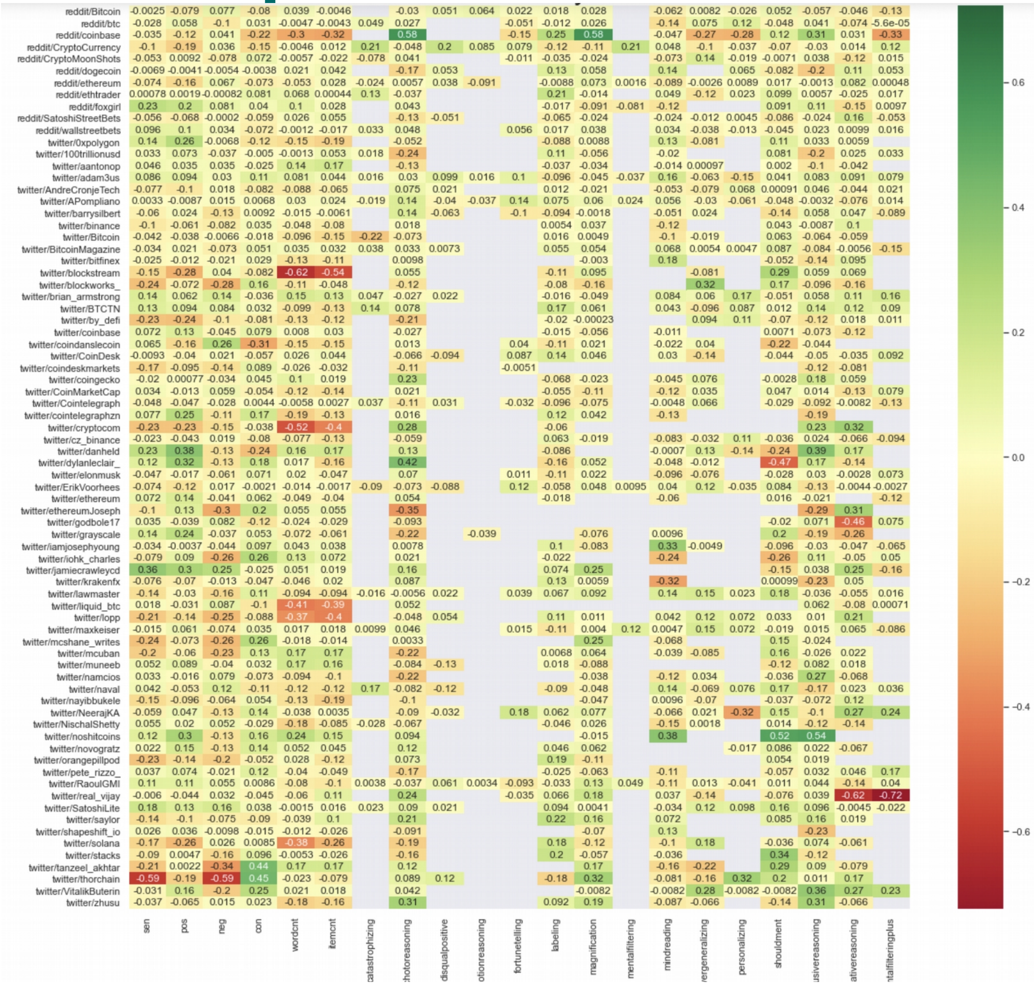

## Blended sentiment only

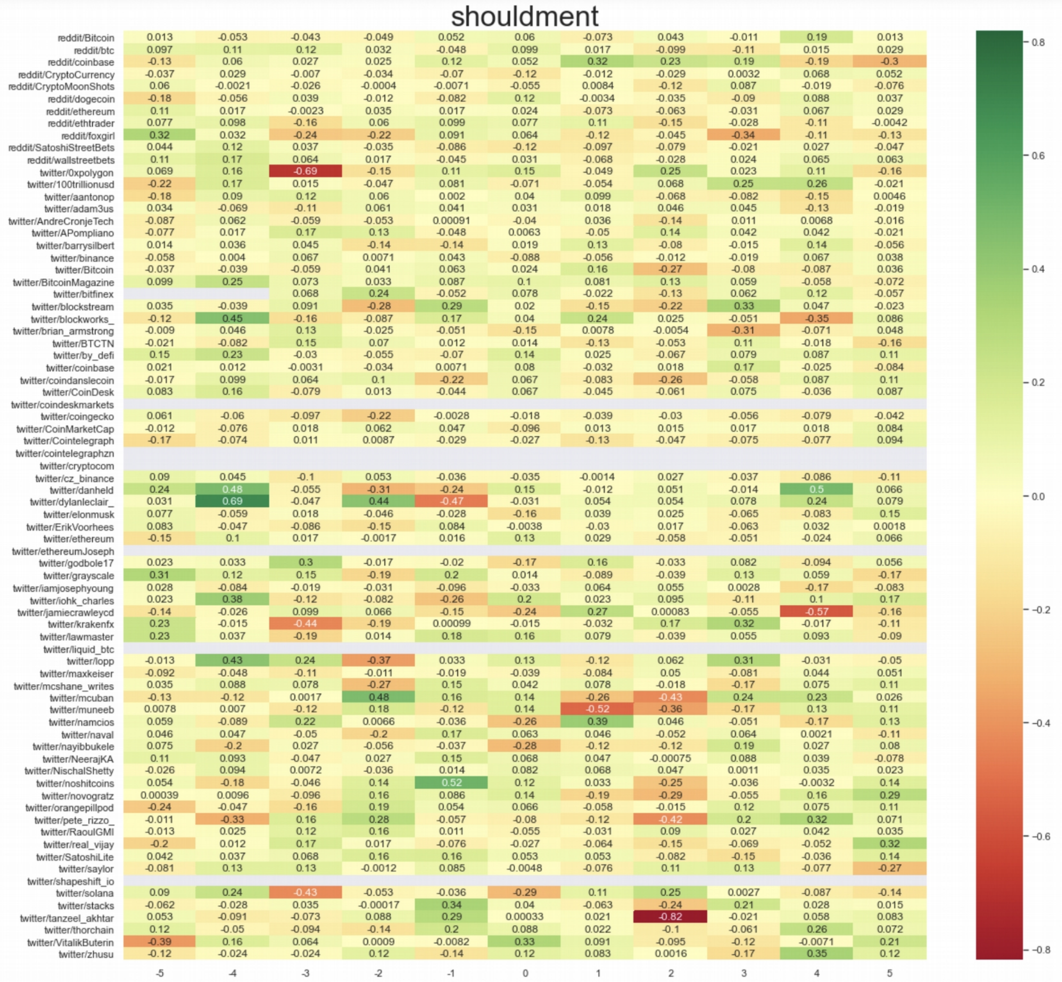



## Blended sentiment & CBS



# Temporal "Cube" Correlational Analysis



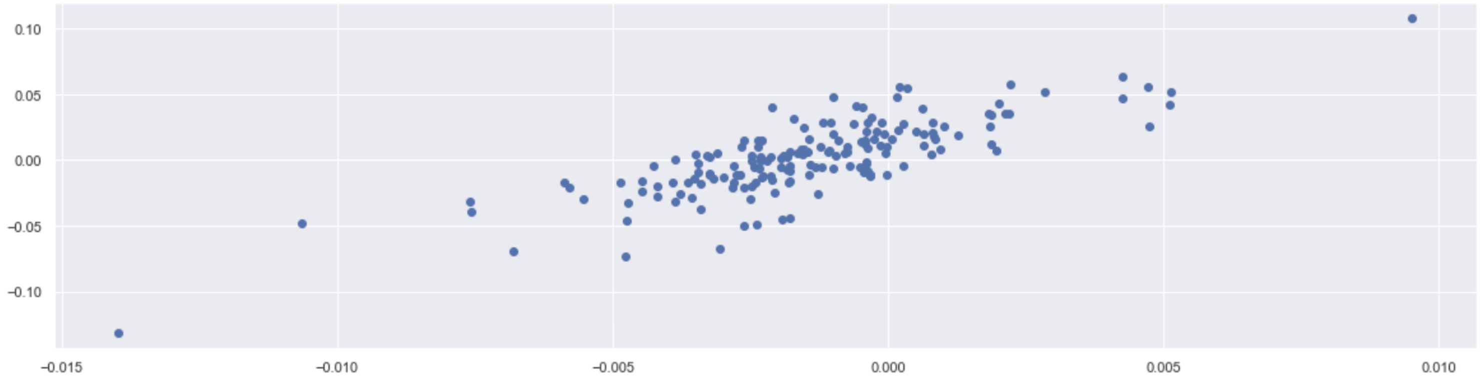


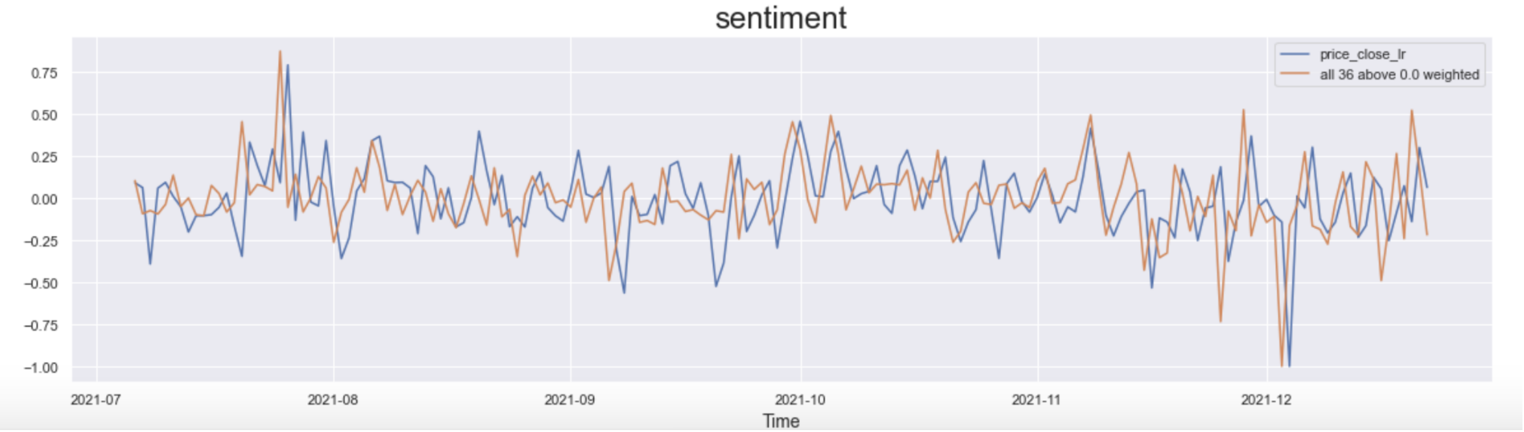
# Building Compound Indicator


```
140 140 30 30 27 0.8030700844490558 0.1081875666717884
```

# Correlating Price LR/Diff with CI

```
plt.scatter(blended, price)
plt.show()
plot_series([blended],None,'sentiment',resampled_price_df,'price_close_lr',norm=True)
plot_series([pred],None,'price prediction',resampled_price_df,'price_close_lr',norm=False)
```



# Predicting Price by Diff


```
140 140 30 30 12 0.654974010649093 0.3141163305338772
```

# Predicting Price by Diff on Intervals


```
10 110 120 0.6592236094660446 -0.211605336295256 35
```


```
20 120 130 0.6980325501824558 0.18497522819543935
```

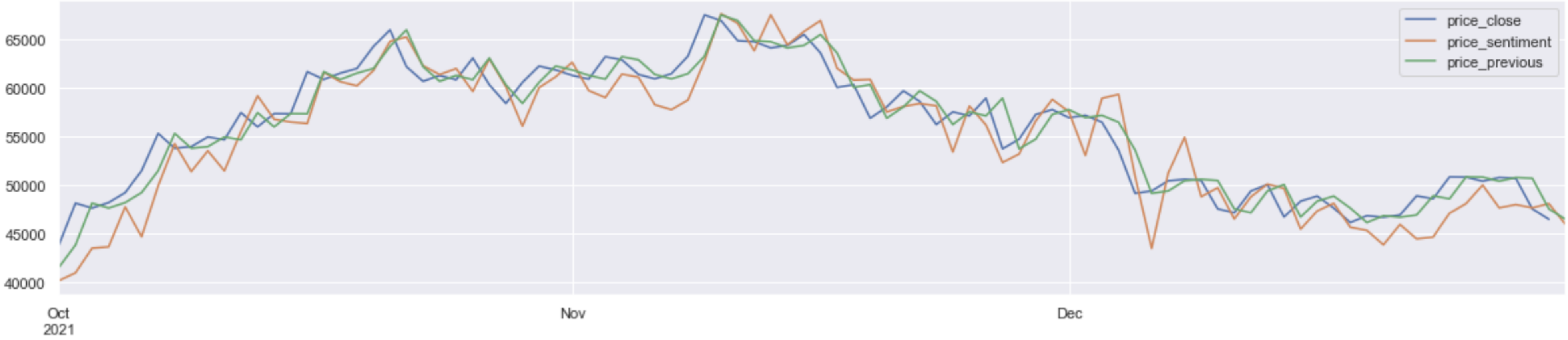
# Daily Price Prediction by CI

```
price90_df = stsql.get_ohlcvs_range_df(from_time,till_time,period=3600*24)
df = pd.concat([predicted90_df, price90_df], axis=1)

mape = mape_vectorized_v2(df['price_close'],df['price_previous'])
da = safe_mean_directinal_accuracy(df['price_close'],df['price_previous'])
print('price_previous',mape,da)
mape = mape_vectorized_v2(df['price_close'],df['price_sentiment'])
da = safe_mean_directinal_accuracy(df['price_close'],df['price_sentiment'])
print('price_sentiment',mape,da)

p = df[['price_close','price_sentiment','price_previous']].plot()
```

```
price_previous 0.026243467195859828 0.43333333333333335
price_sentiment 0.04143159097794829 0.47777777777777778
```



# Hourly Price Prediction by CI

```
predicted30h_df = pd.DataFrame(prices30h,columns=['price_sentiment','price_previous','diff_predicted','time'])
predicted30h_df.set_index('time',inplace=True)

price30h_df = stsql.get_ohlcvs_range_df(from_time,till_time+dt.timedelta(hours=1),period=3600)
print(len(predicted30h_df))
assert len(predicted30h_df) == len(price30h_df)
df = pd.concat([predicted30h_df, price30h_df], axis=1)

mape = mape_vectorized_v2(df['price_close'],df['price_previous'])
da = safe_mean_directinal_accuracy(df['price_close'],df['price_previous'])
print('price_previous',mape,da)
mape = mape_vectorized_v2(df['price_close'],df['price_sentiment'])
da = safe_mean_directinal_accuracy(df['price_close'],df['price_sentiment'])
print('price_sentiment',mape,da)

p = df[['price_close','price_sentiment','price_previous']].plot()
p = df[:500][['price_close','price_sentiment','price_previous']].plot()
p = df[500:1000][['price_close','price_sentiment','price_previous']].plot()
p = df[1000:1500][['price_close','price_sentiment','price_previous']].plot()
p = df[1500:][['price_close','price_sentiment','price_previous']].plot()
```

```
2185
price_previous 0.004637059941957259 0.4739010989010989
price_sentiment 0.020060797850708 63 0.4958791208791209
```

# TODO Next?

Overfitting–tolerant blending?
Manual channel grouping by clustering for cleaner blending?
Add account for item counts and text volumes?
Cleaner normalization based on text volumes?
Hourly training on daily intervals?
Using other ML approaches to “the media cube” data?
Currencies other than Bitcoin?

# Thank you and welcome!

Anton Kolonin
akolonin@aigents.com
Facebook: akolonin
Telegram: akolonin

autonio.foundation

SingularityNET