FINAL PITCH

GROUP PROJECT x EUKLID

TODAY'S AGENDA

03.05.2024

I. PROJECT DESCRIPTION
II. FEATURE ENGINEERING
III. BENCHMARKS
IV. FINAL MODEL
V. CONCLUSIONS

PROJECT DESCRIPTION

OUR CHALLENGE

Design and develop an AI-driven systematic trading model, built on TA indicators, that can navigate the volatile financial markets with precision and agility.

OUR DATA

Financial data consisting of historical market prices and volumes from companies (IBM, Microsoft, Amazon) and indexes (CAC40, S&P 500, NASDAQ).

LIBRARIES

OUR DATA

- PyTorch
- Keras
- Numpy
- Matplotlib.pyplot
- Pandas
- Pandas_ta
- Hurst

PREPROCESSING AND FEATURE ENGINEERING

- MISSING VALUES
- SPLITS ADJUSTMENTS
- CLOSE PRICE ADJUSTMENTS
- STANDARDIZATION
- INDICATORS

PREPROCESSING

MISSING VALUES

STOCK
e.g. Amazon:
Fillna using Yahoo Finance
(week 2022-09-18)

INDEX
e.g. S&P 500:
Fillna using Yahoo Finance
(weeks 2023-05-14 and 2023-05-21)

PREPROCESSING

SPLITS ADJUSTMENTS

STOCK

e.g. Amazon:

- 1998-06-02 2:1
- 1999-01-05 3:1
- 1999-09-05 2:1
- 2022-06-06 20:1

INDEX

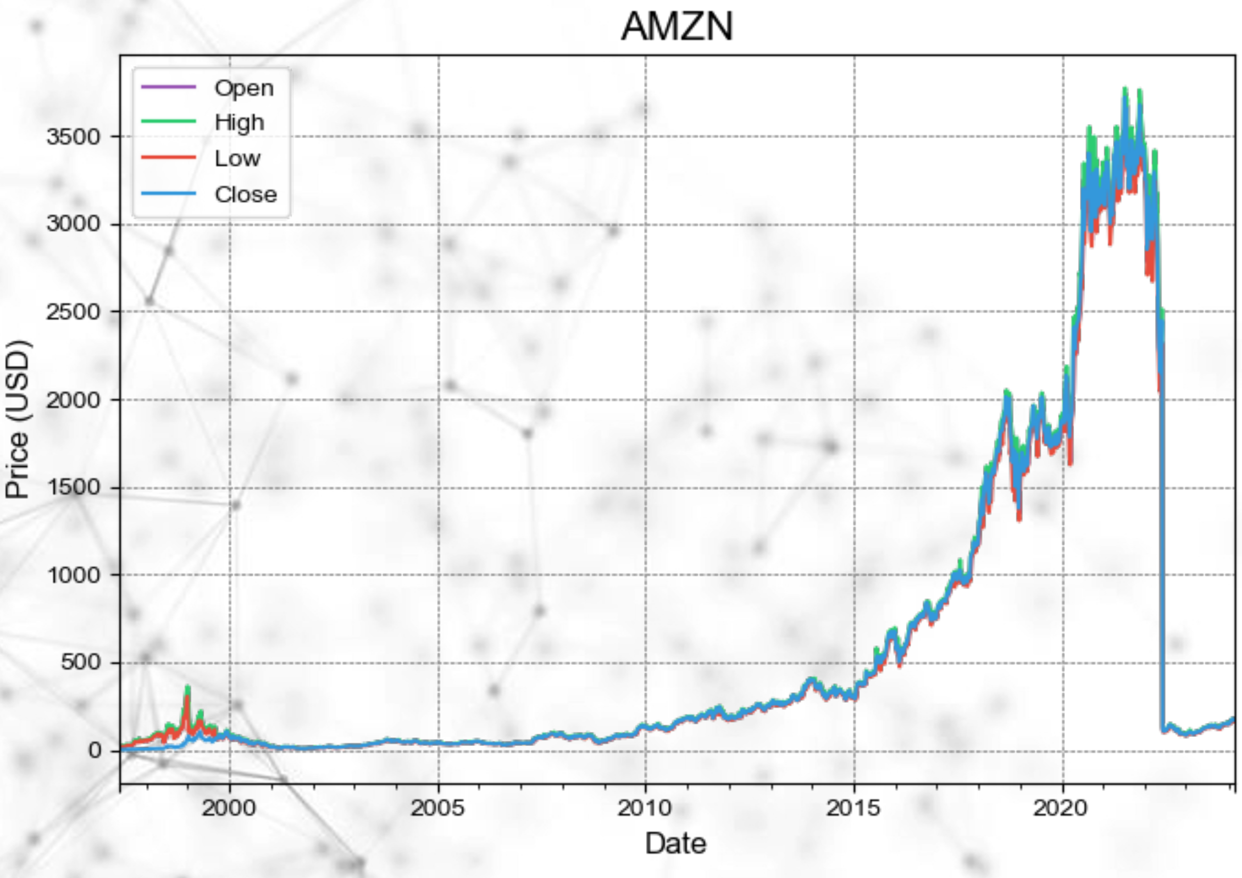
Capitalization-Weighted Indexes are based on Market Capitalization, and as such, aren't affected by stock splits.

PREPROCESSING

PRICE ADJUSTMENT

STOCK
e.g. Amazon:

- 1998-06-02 2:1
- 1999-01-05 3:1
- 1999-09-05 2:1
- 2022-06-06 20:1

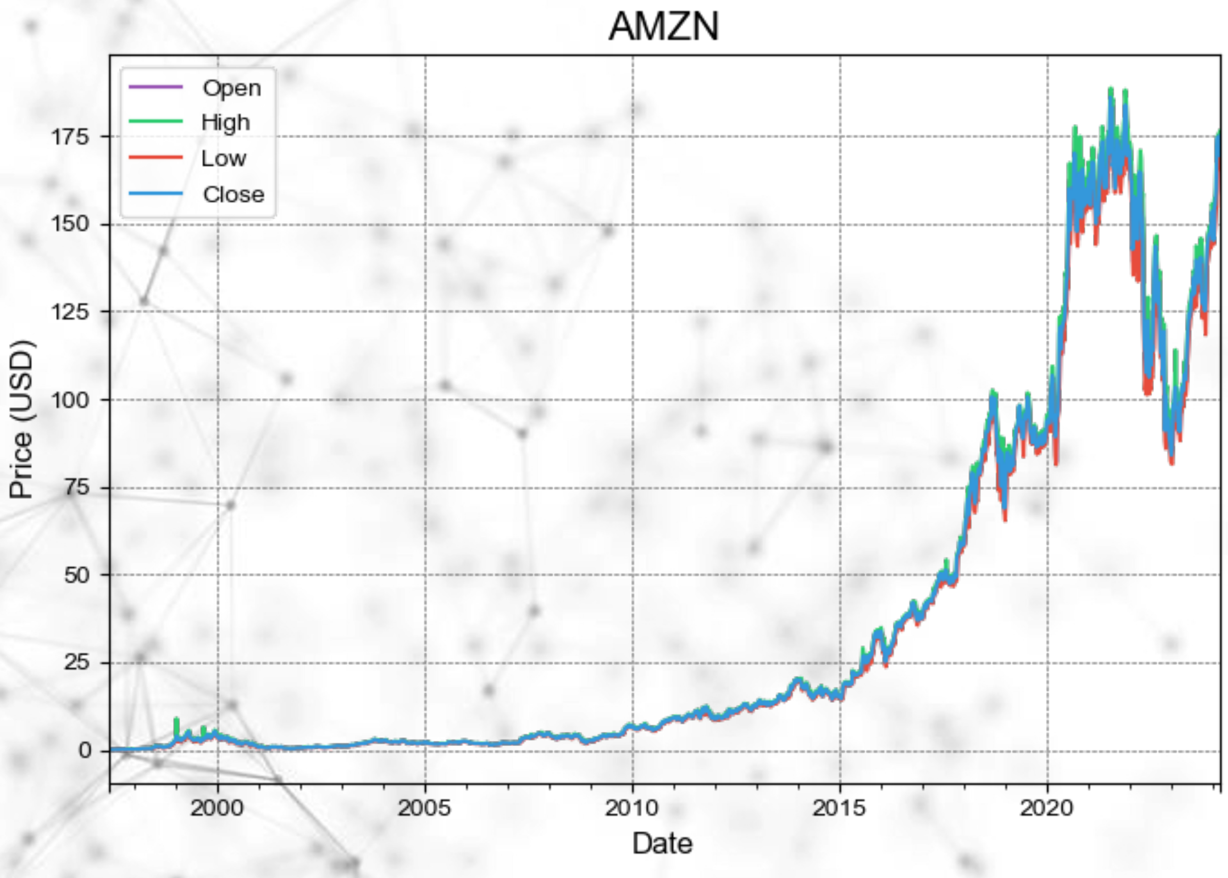


FEATURE ENGINEERING

PRICE ADJUSTMENT

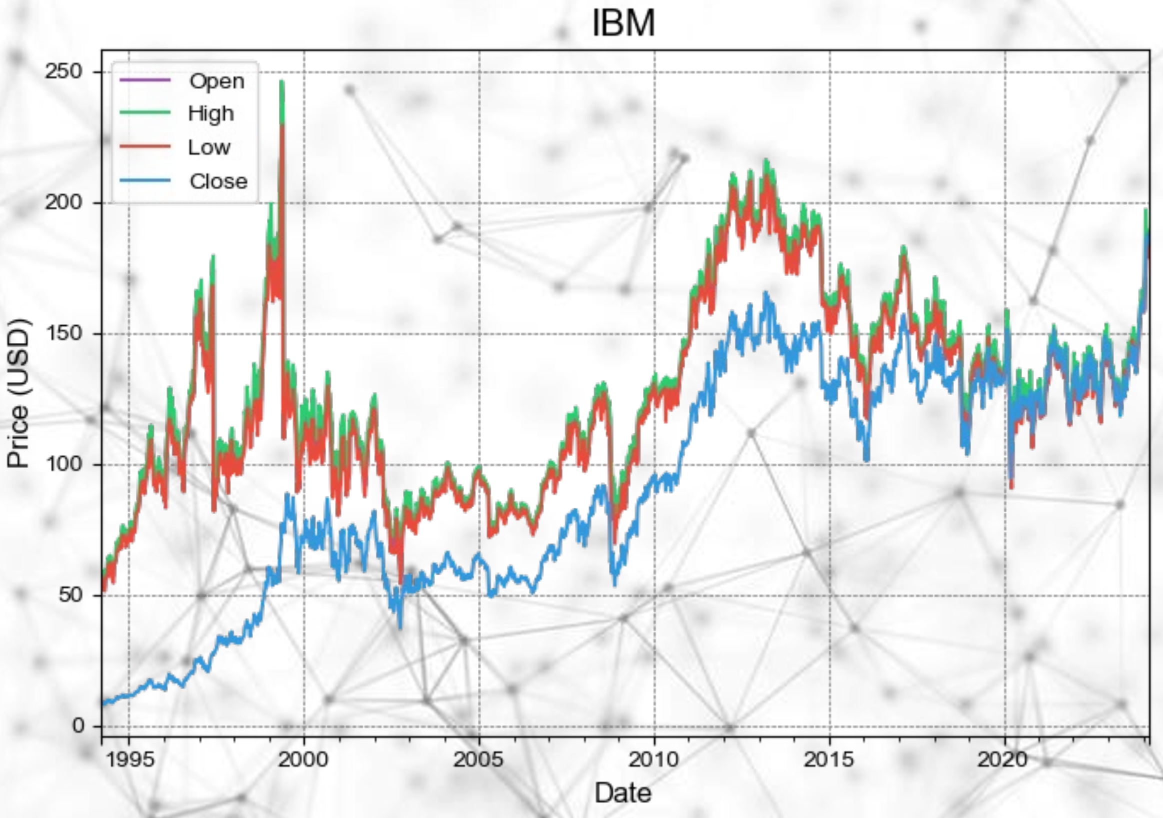
STOCK
e.g. Amazon:

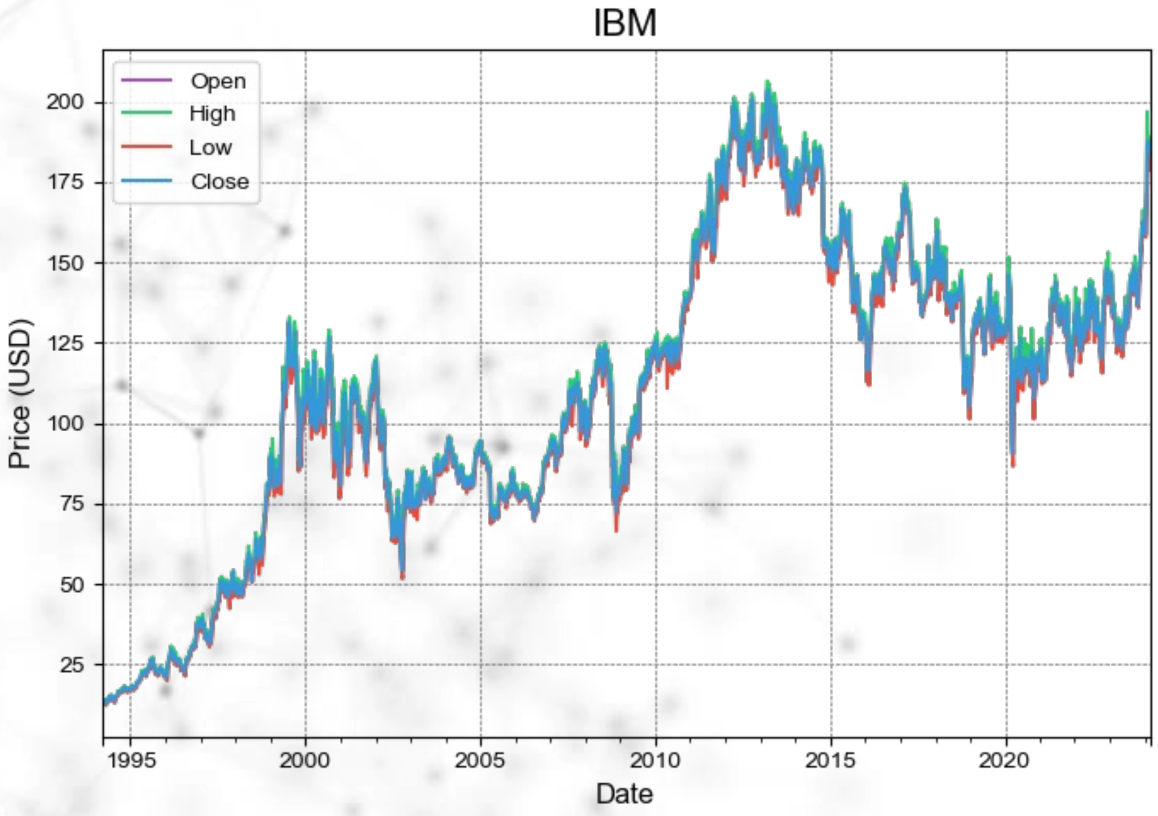
- 1998-06-02 2:1
- 1999-01-05 3:1
- 1999-09-05 2:1
- 2022-06-06 20:1



FEATURE ENGINEERING

CLOSE PRICE ADJUSTMENT

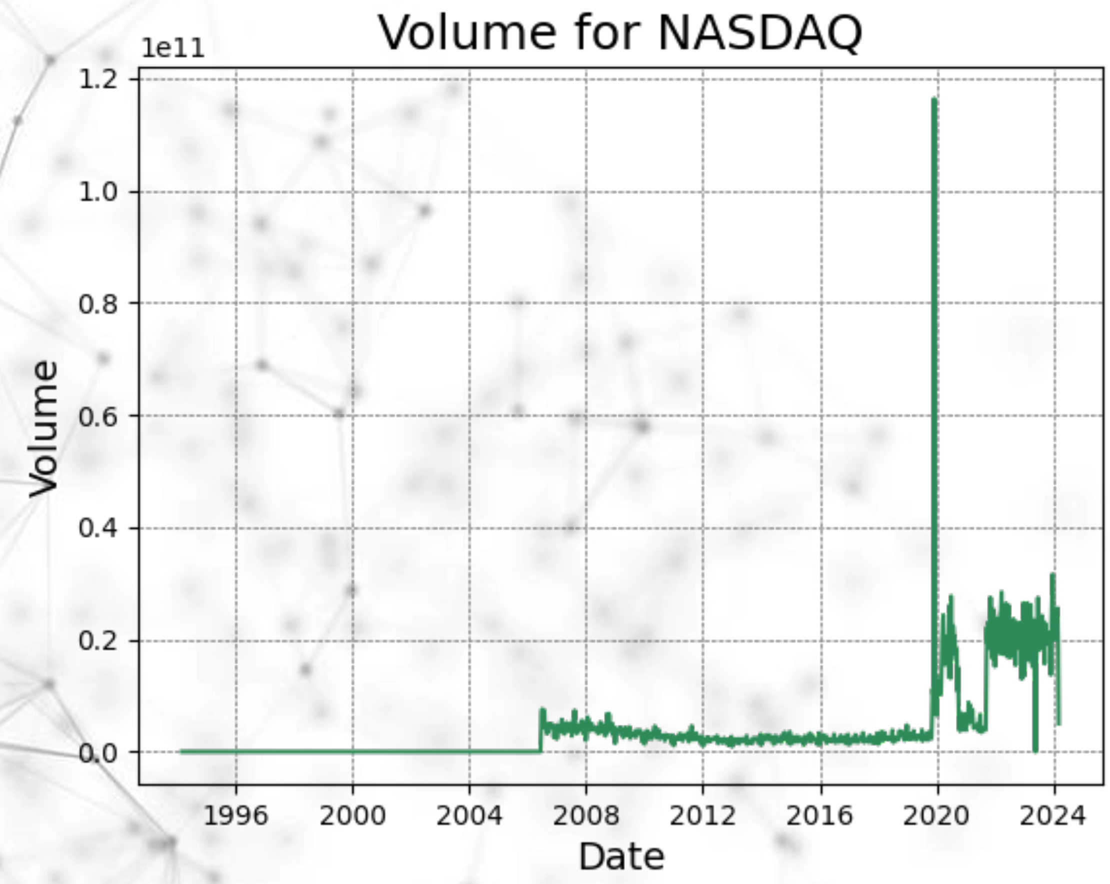




FEATURE ENGINEERING

STANDARDIZATION

- StandardScaler for prices: highly recommended
- MinMaxScaler for volumes, considering that there were long period in which volumes were recorded to be zero



FEATURE ENGINEERING

INDICATORS

RSI

- Relative Strength Index
- Momentum Oscillator
- Measures the size and speed of price changes

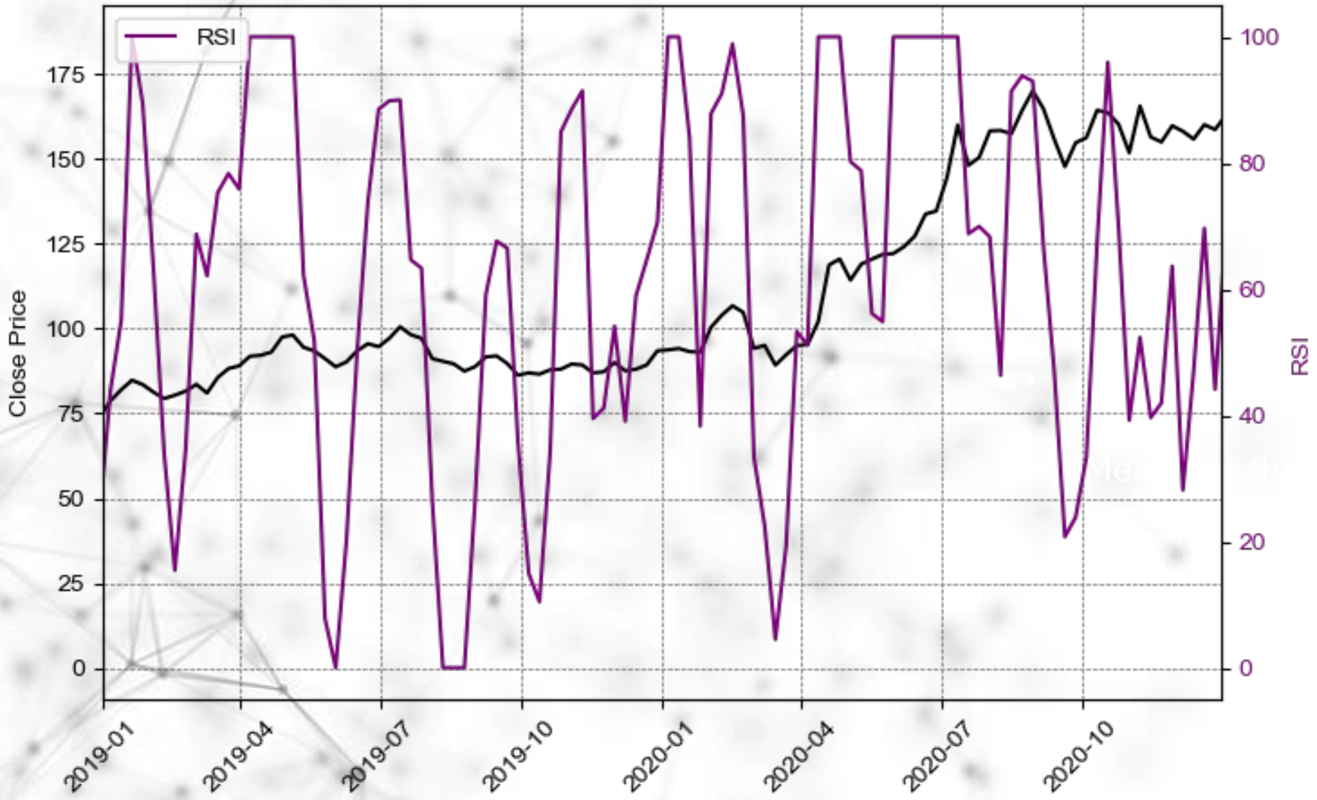
$$RSI = 100 - \frac{100}{1 + \left(\frac{Average\ Gain}{Average\ Loss}\right)}$$

FEATURE ENGINEERING

INDICATORS

RSI

- Relative Strength Index
- Momentum Oscillator
- Measures the size and speed of price changes



FEATURE ENGINEERING

INDICATORS

EMA

- Exponential Moving Average
- Reacts more significantly to recent price changes

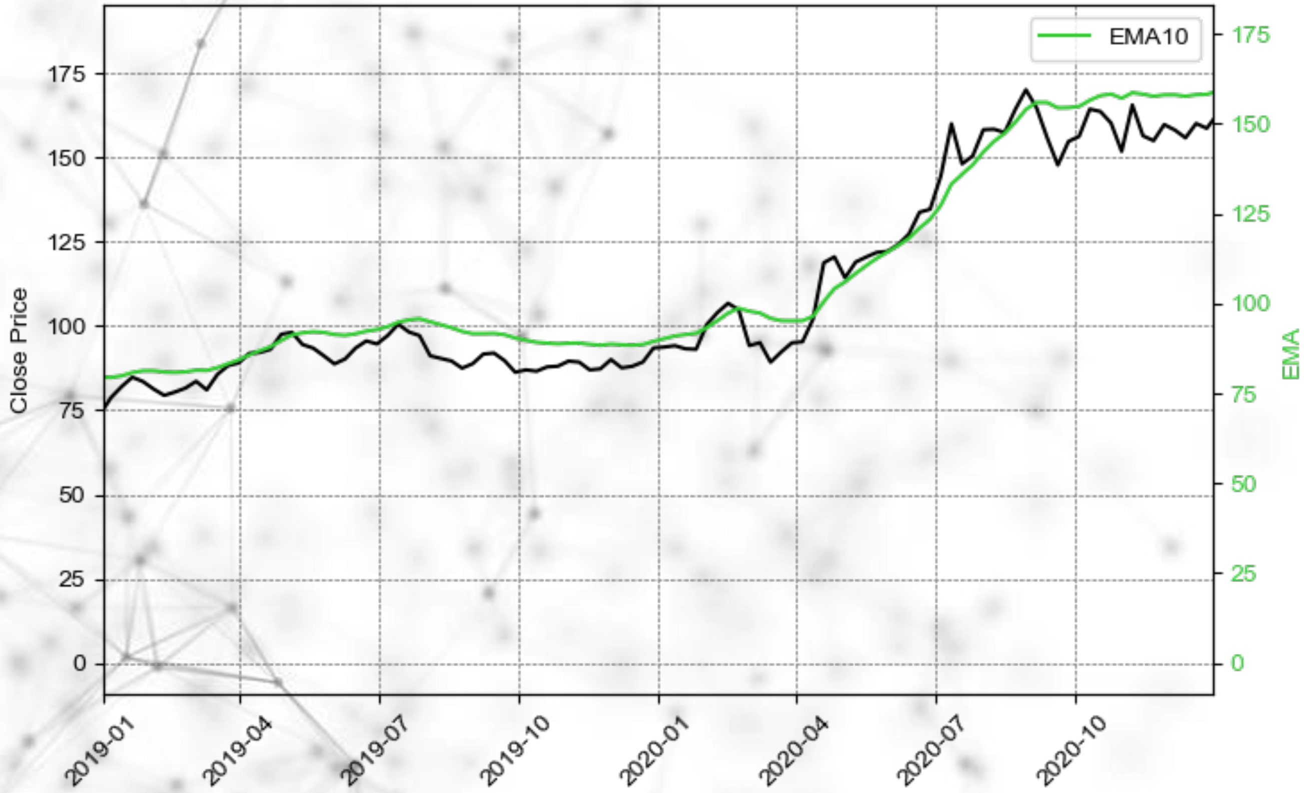
$$EMA = \left(Value_{Today} * \left(\frac{Smoothing}{1 + Days}\right)\right) + $$
$$+ EMA_{Yesterday} * (1 - (\frac{Smoothing}{1 + Days}))$$

FEATURE ENGINEERING

INDICATORS

EMA

- Exponential Moving Average
- Reacts more significantly to recent price changes



FEATURE ENGINEERING

INDICATORS

CPC

- Current Price Change
- Used to measure recent stock price movements

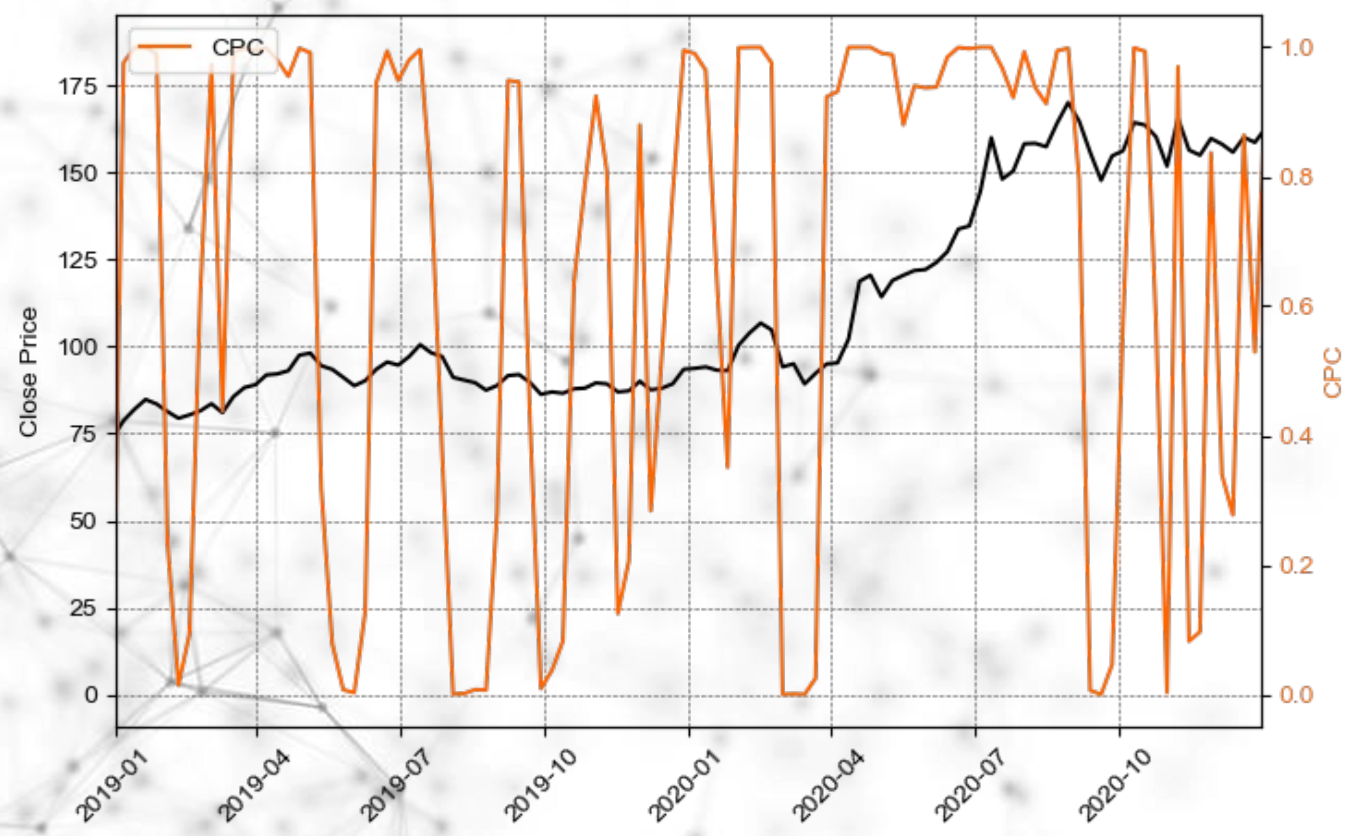
$$CPC = \frac{1}{1 + e^{-\left(\frac{C_t - MA_{t-1,t-i}}{MA_{t-1,t-i}} * 100\right)}}$$

FEATURE ENGINEERING

INDICATORS

CPC

- Current Price Change
- Used to measure recent stock price movements



FEATURE ENGINEERING

INDICATORS

CCI

- Commodity Channel Index
- Momentum Oscillator
- Measures a stocks' variation from the statistical mean

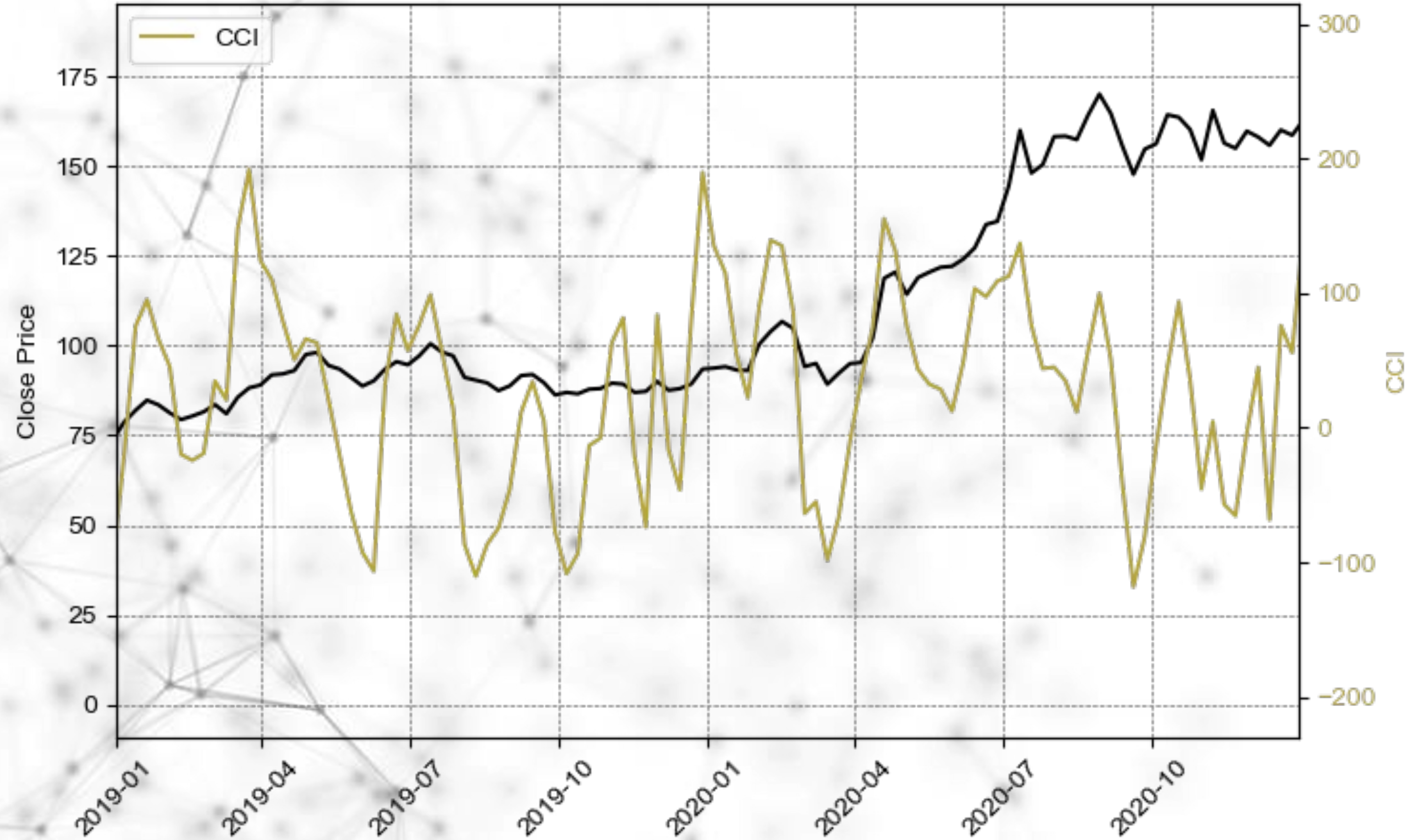
$$CCI = \frac{M_t - SM_t}{0.015 * D_t}$$

FEATURE ENGINEERING

INDICATORS

CCI

- Commodity Channel Index
- Momentum Oscillator
- Measures a stocks' variation from the statistical mean



FEATURE ENGINEERING

INDICATORS

EMADN

- Normalized Exponential Moving Average
- Simple way of interpreting ”Golden Cross” & “Dead Cross”

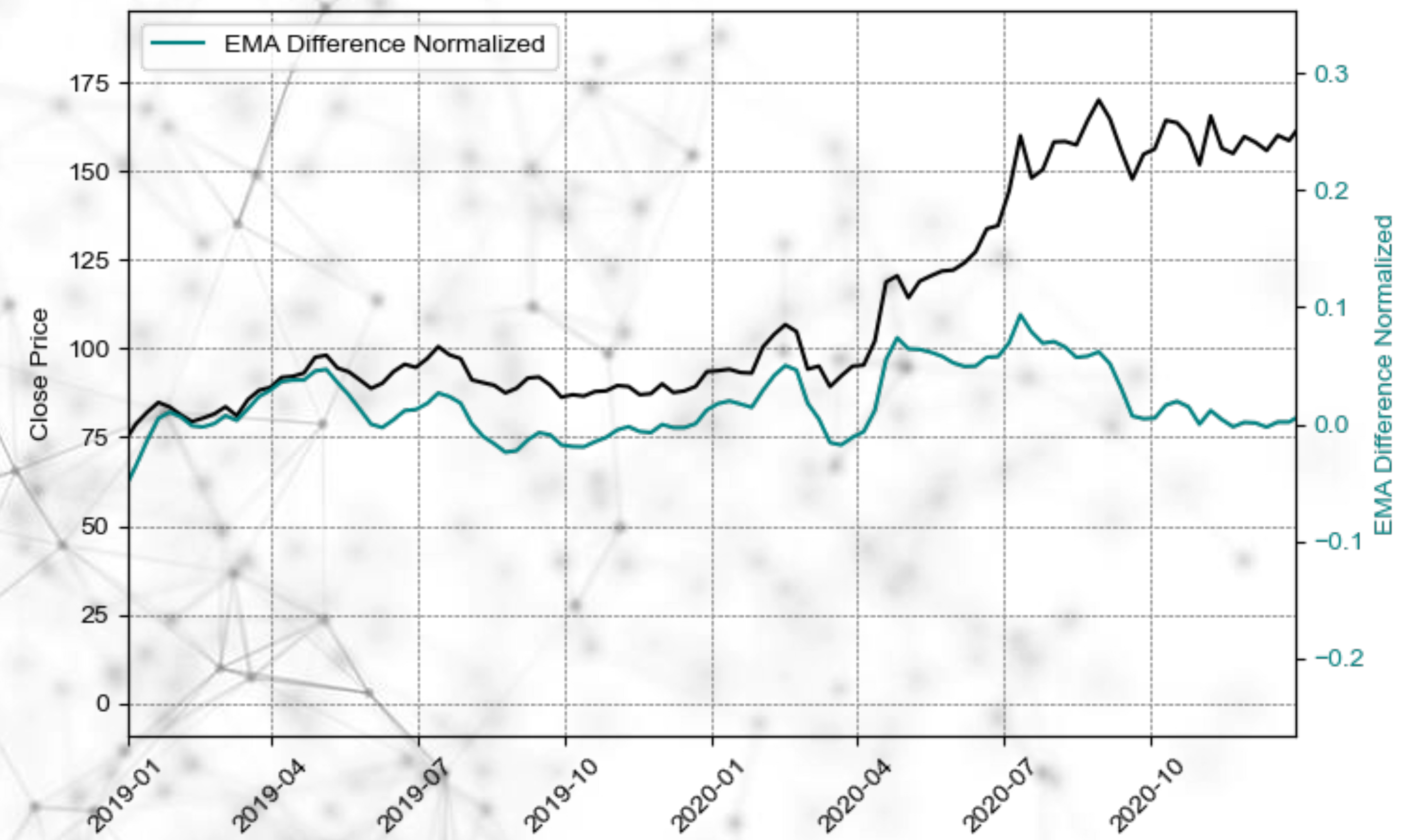
$$EMADN = \frac{EMA_{SHORT} - EMA_{LONG}}{EMA_{LONG}}$$

FEATURE ENGINEERING

INDICATORS

EMADN

- Normalized Difference of Exponential Moving Averages
- Simple way of interpreting ”Golden Cross” & “Dead Cross”



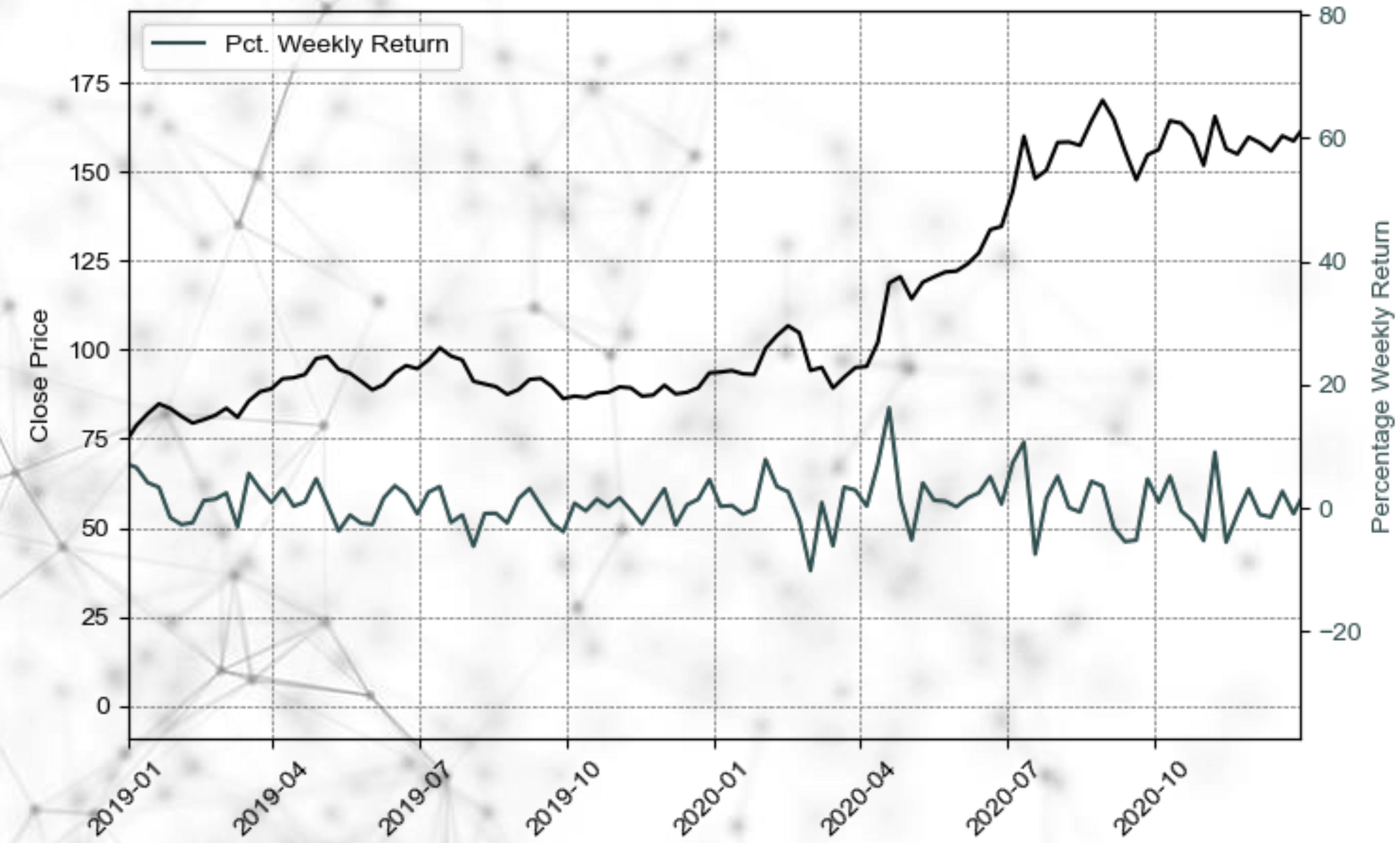
FEATURE ENGINEERING

OTHER FEATURES

Weekly Return

- Difference in Close between each week and the previous, normalized

$$\frac{C(t) - C(t-1)}{C(t-1)}$$



BENCHMARKS

ARIMA

AutoRegression
Integrated
Moving Average

LSTM

Long
Short
Term
Memory

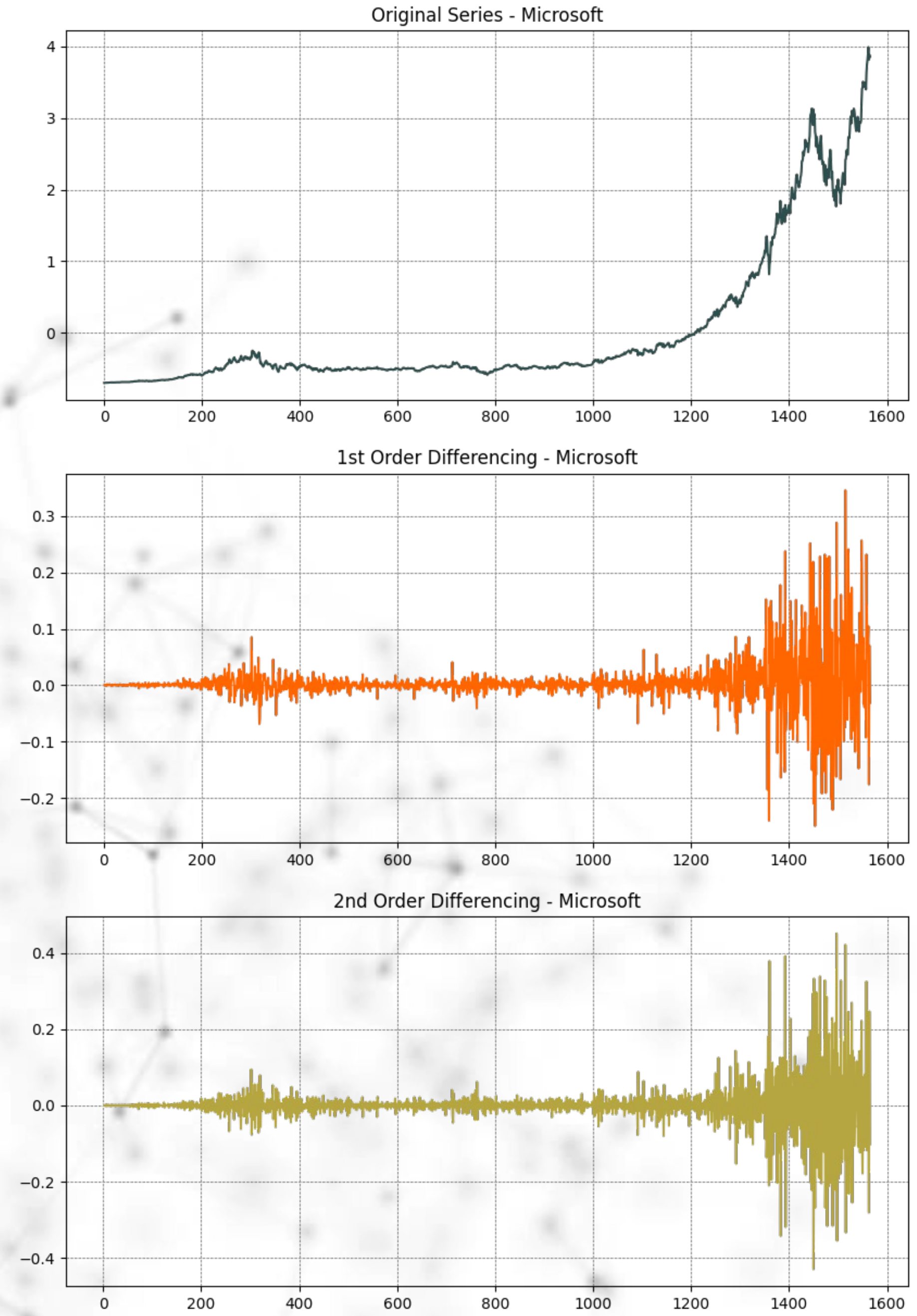
ARIMA

$$Y_t = c + \emptyset_1 Y_{t-1} + \emptyset_2 Y_{t-2} + \ldots + \emptyset_p Y_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t$$

- d is the number of nonseasonal differences needed for stationarity
- p is the number of autoregressive terms
- q is the number of lagged forecast errors in the prediction equation
- Yt is the value of the time series at time tt.
- c is a constant term.
- $\emptyset_j$ are the parameters of the autoregressive (AR) part of the model.
- θ_i are the parameters of the moving average (MA) part of the model.

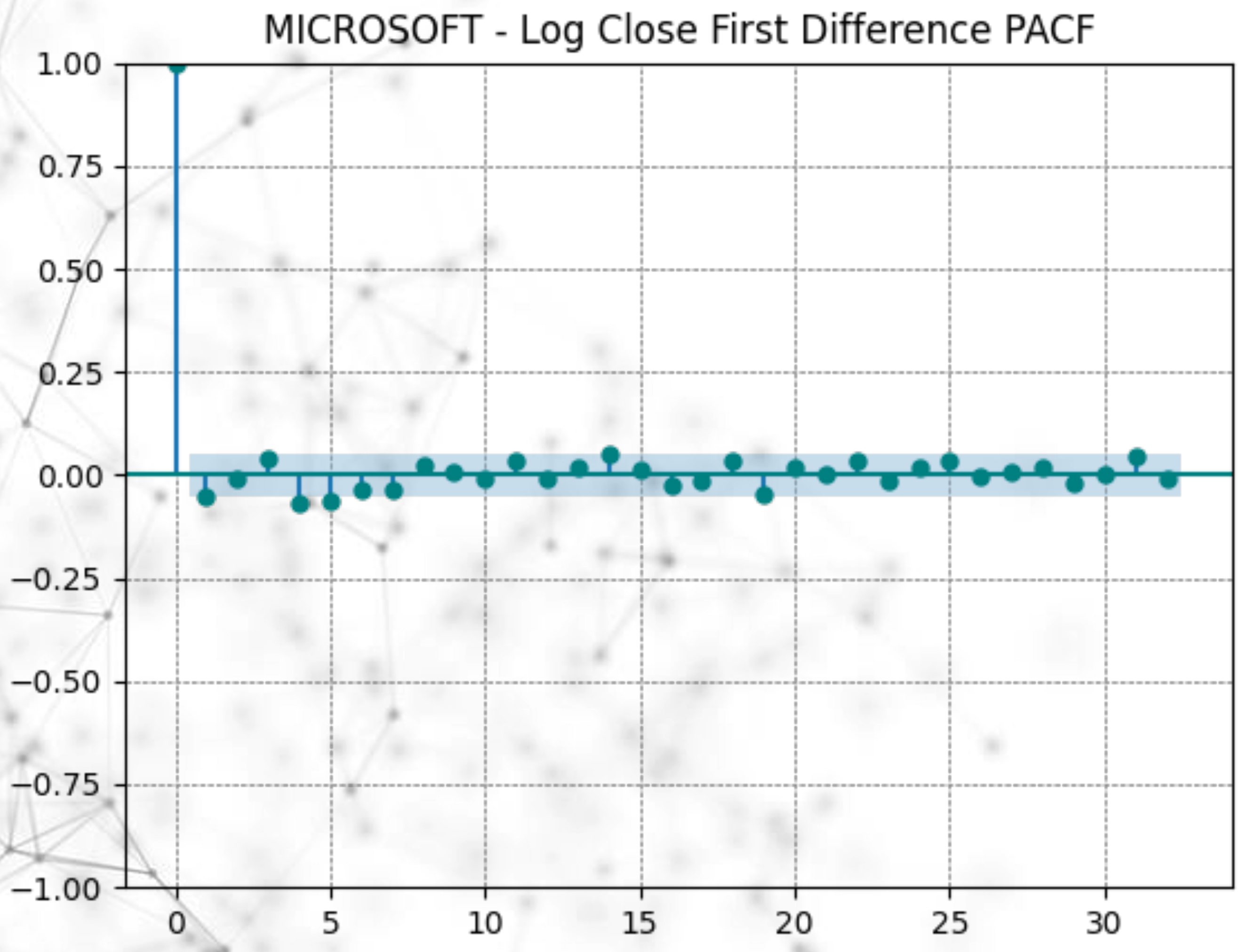
ARIMA

- d is the number of nonseasonal differences needed for stationarity
- p is the number of autoregressive terms
- q is the number of lagged forecast errors in the prediction equation



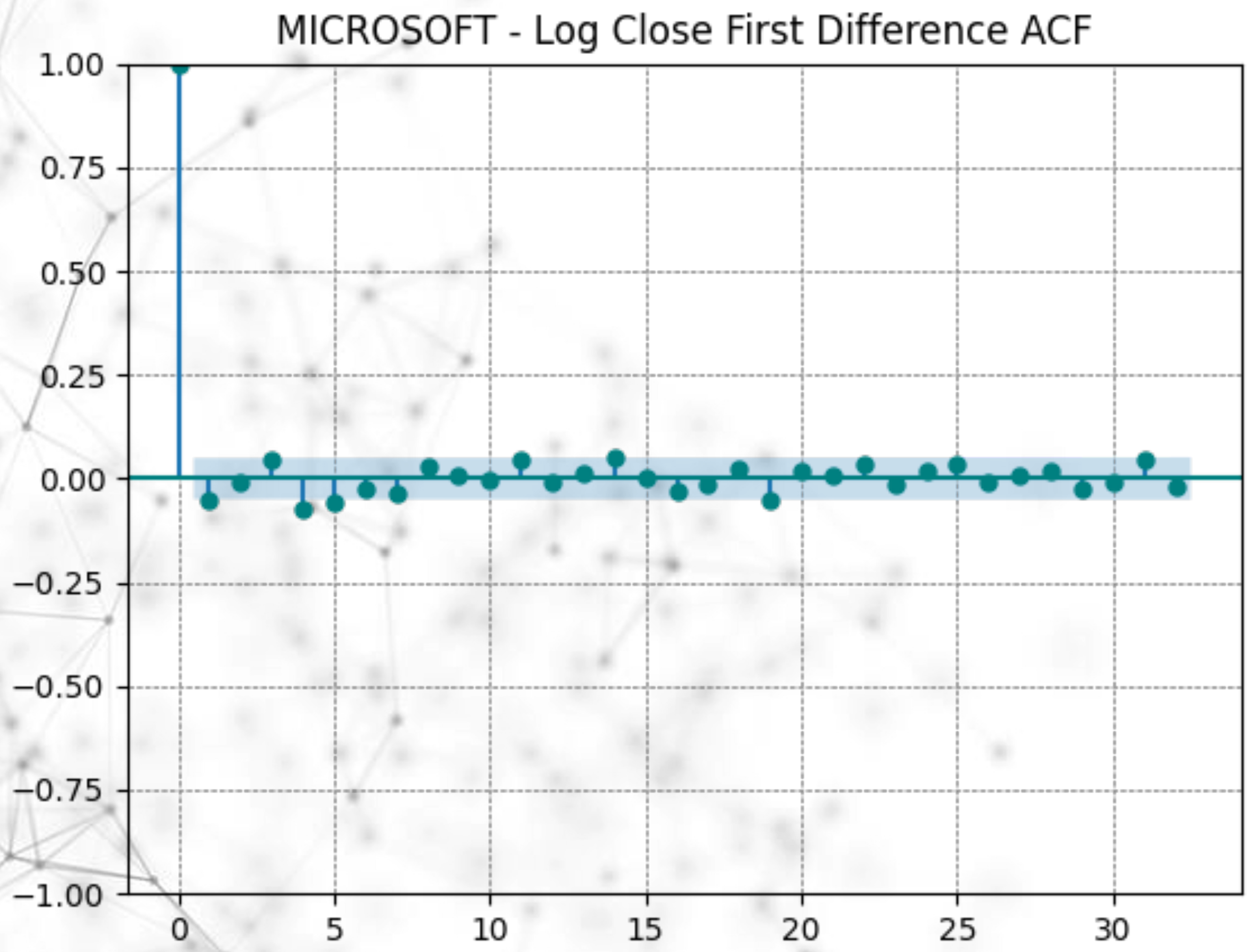
ARIMA

- d is the number of nonseasonal differences needed for stationarity
- p is the number of autoregressive terms
- q is the number of lagged forecast errors in the prediction equation

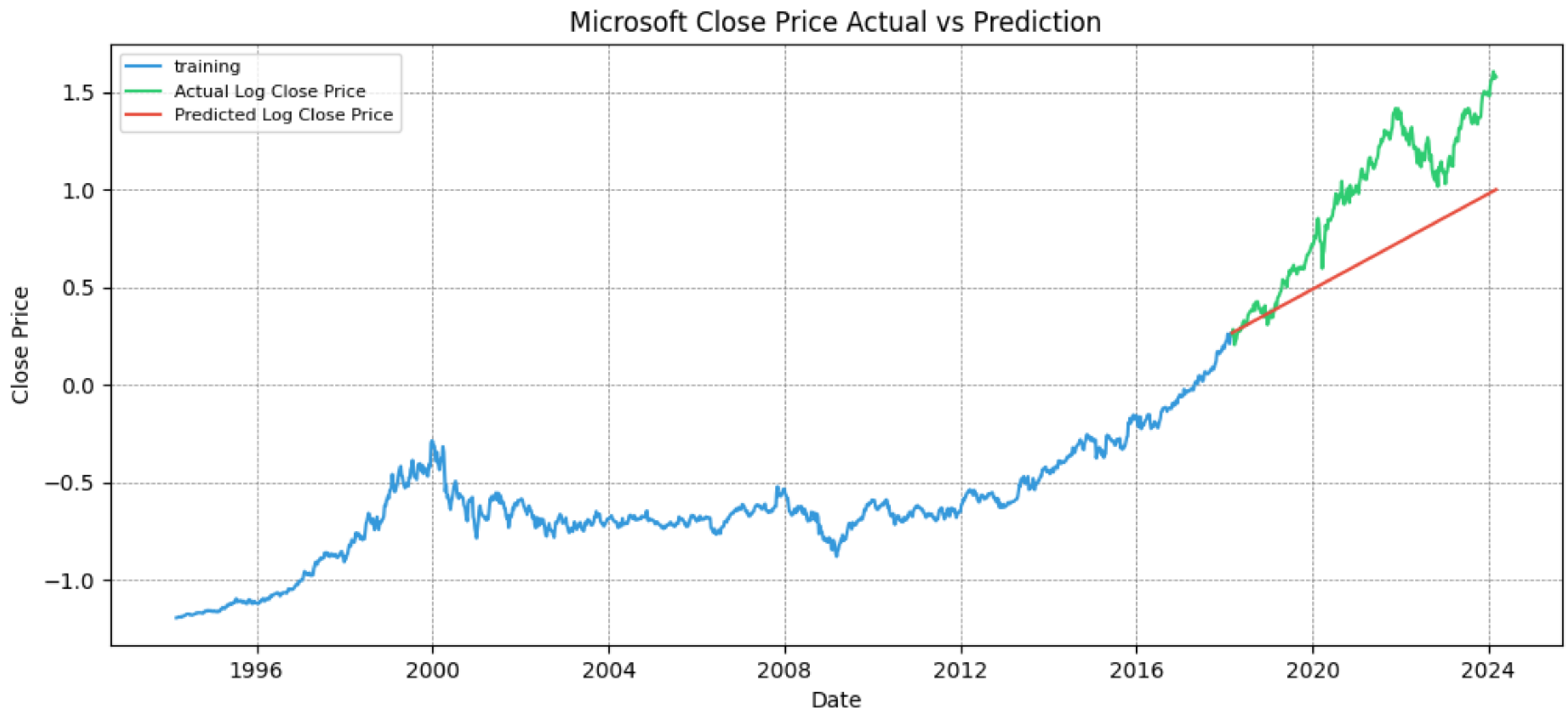


ARIMA

- d is the number of nonseasonal differences needed for stationarity
- p is the number of autoregressive terms
- q is the number of lagged forecast errors in the prediction equation



ARIMA for Microsoft



LSTM

Long
Short
Term
Memory

Which features to include in the window? What are the correct dimensions for the input Tensor?

How many layers? And epochs? And how much do we need to look back to forecast the next?

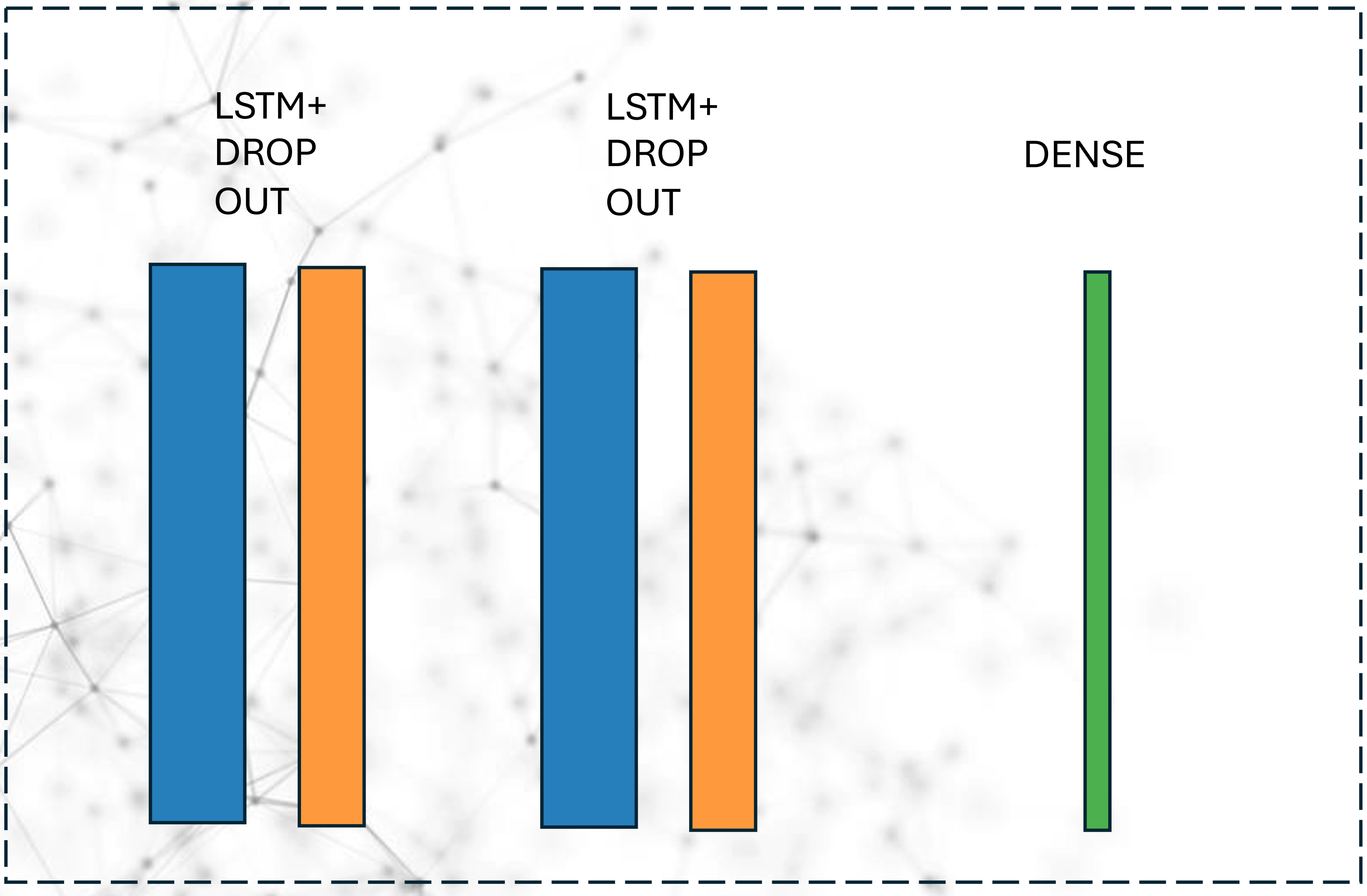
Log return: an opportunity to take?

LSTM MODEL #1

PRICES

INDICATORS: RSI, EMA

PRICES:
WINDOW of 5 weeks

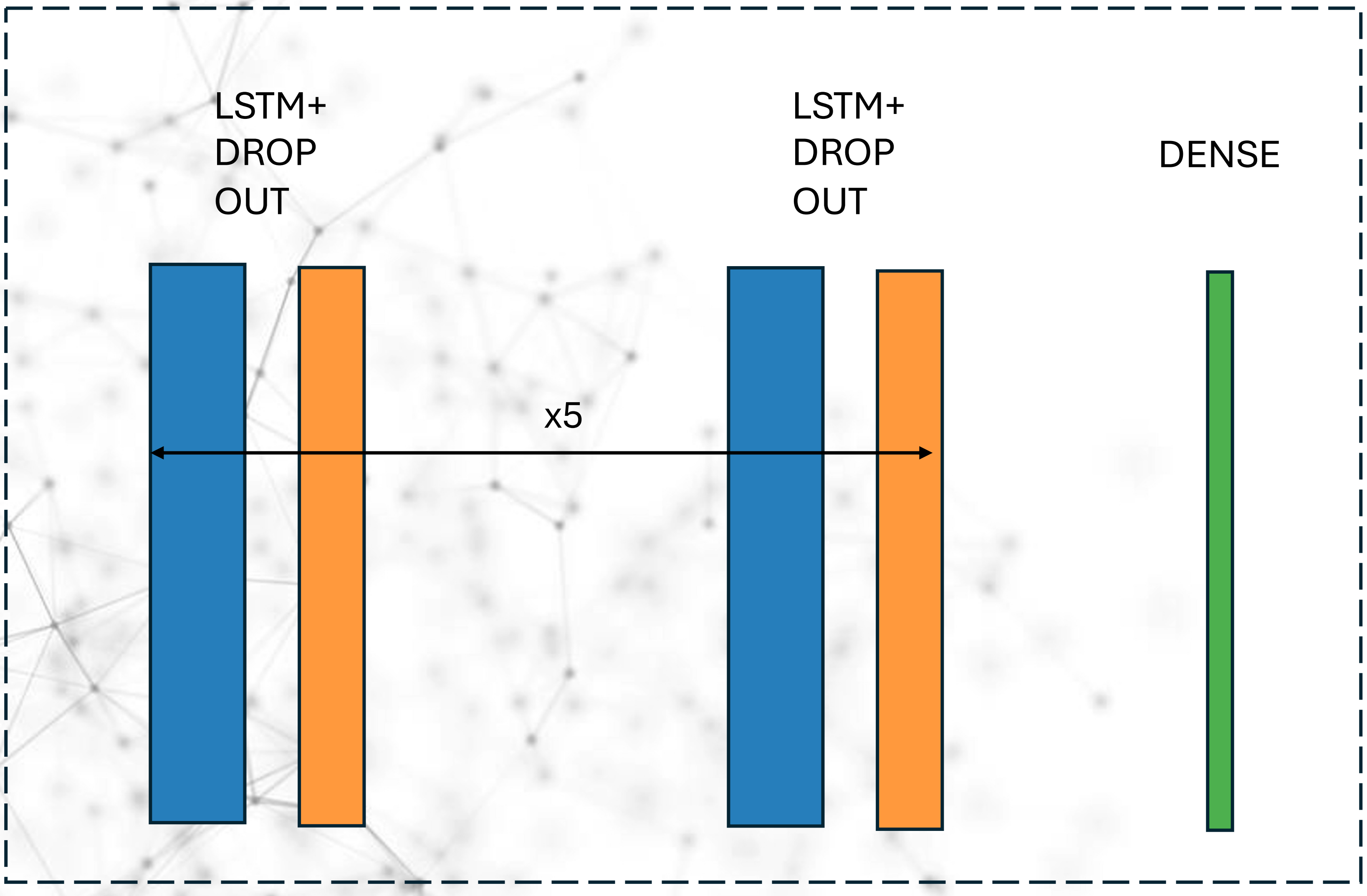


LSTM MODEL #2

PRICES

INDICATORS: RSI, EMA

PRICES and INDICATORS: WINDOW of 10 weeks

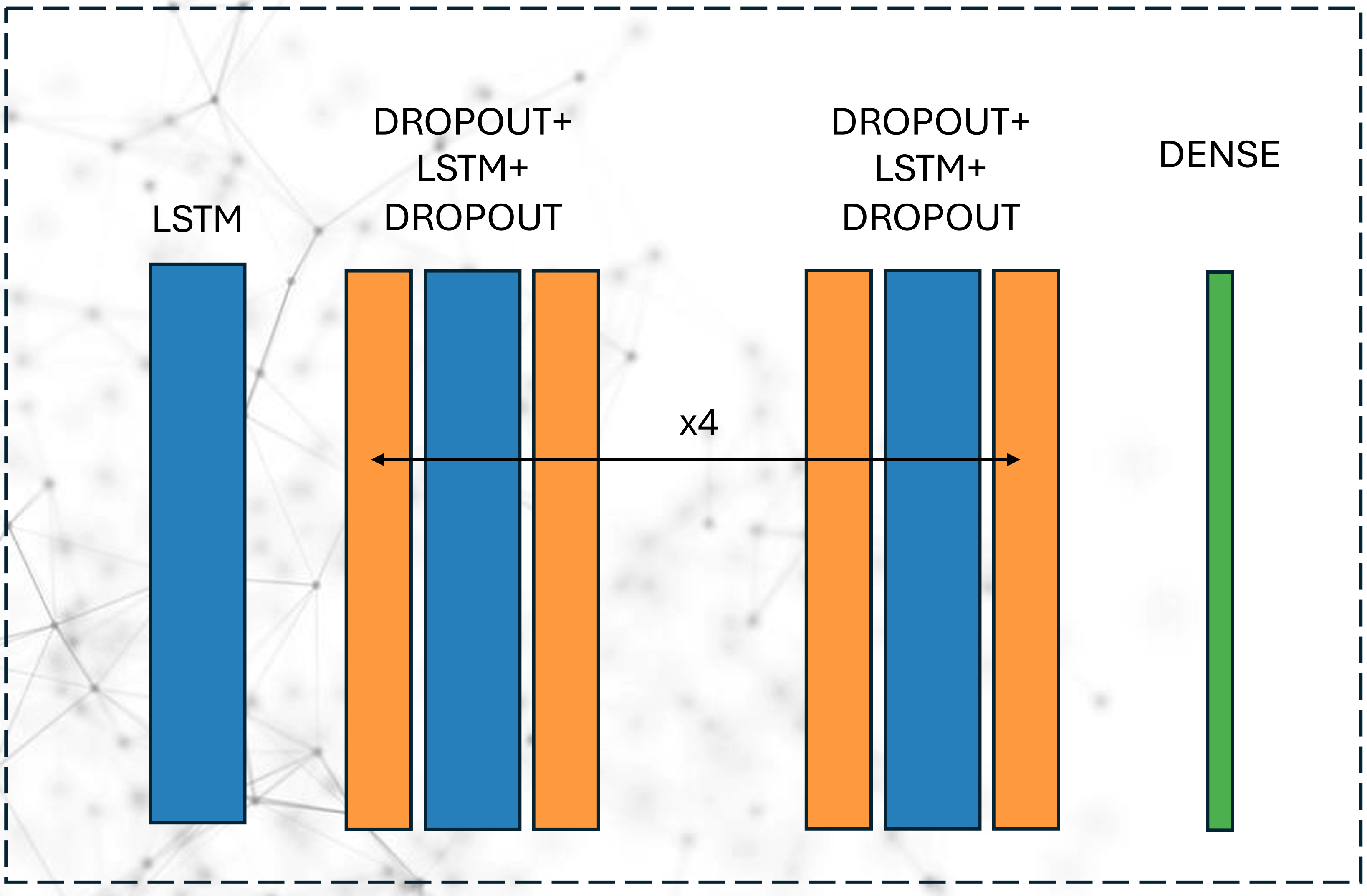


LSTM MODEL #3

PRICES

INDICATORS: RSI, EMA

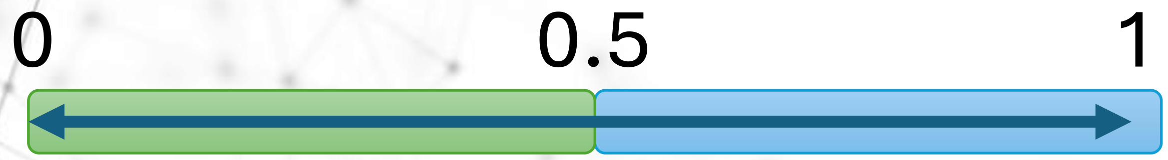
PRICES and INDICATORS:
WINDOW of 10 weeks



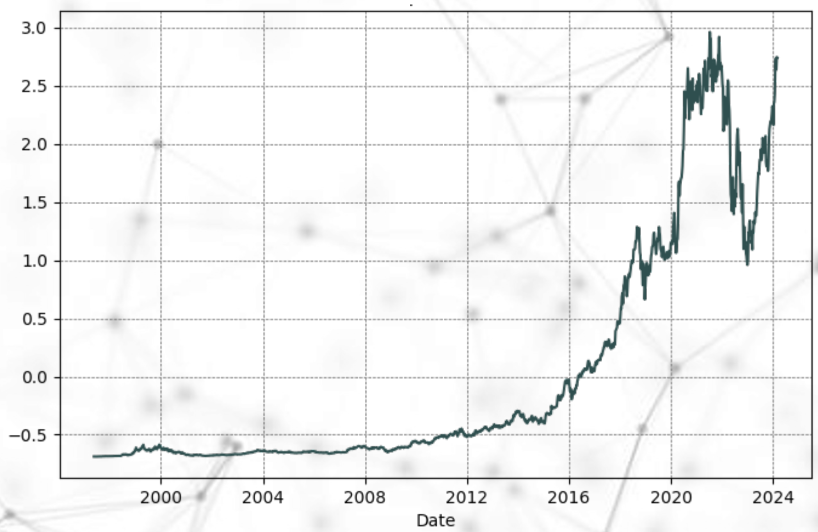
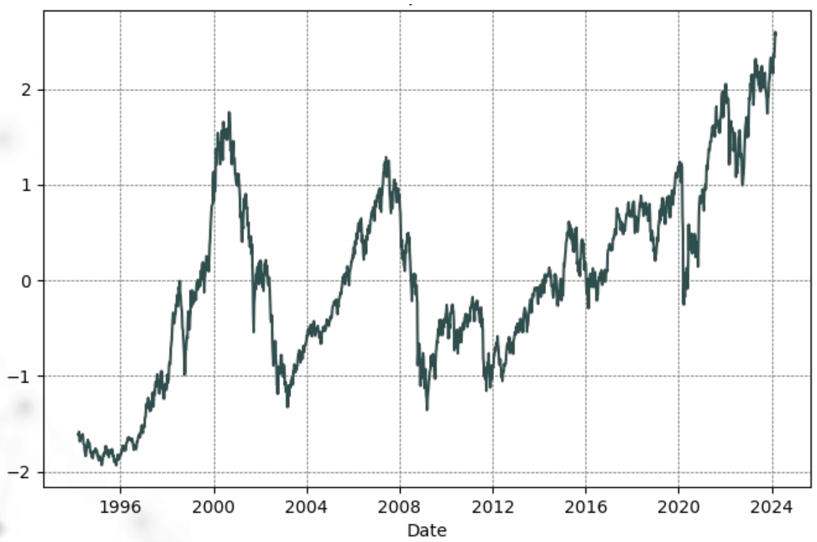
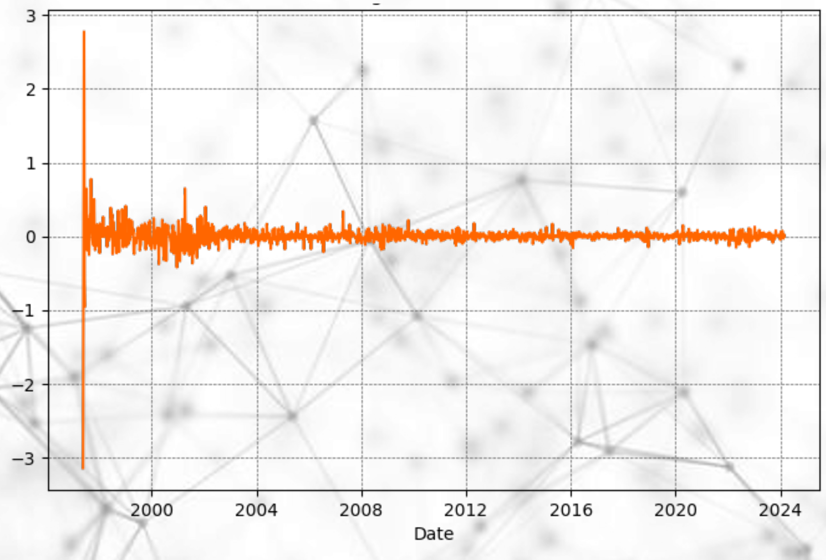
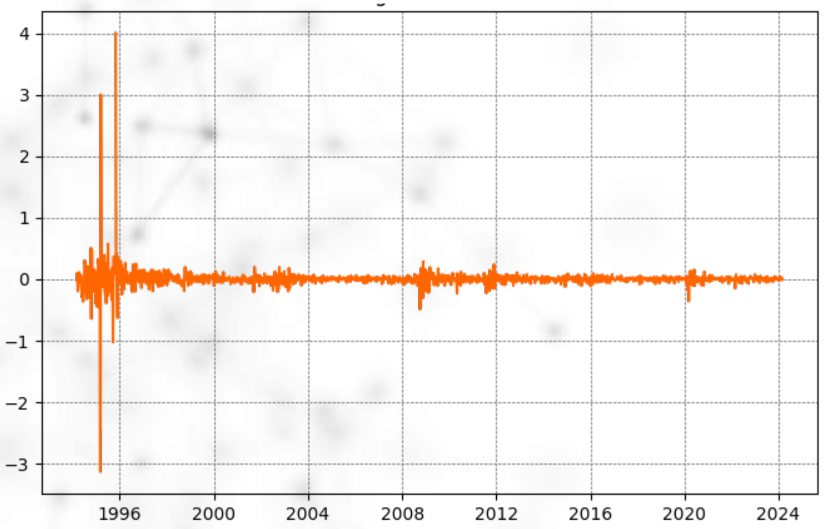
HURST

It’s a statistical measure which is used to study scalilng properties in time series. It measure long term memory of a time series. It uses the variance of a log price series to assess the rate of diffusion behaviour.

Mean reverting - Random walking - Trending



$$E[(\log(t+\tau)-\log(t))^2] \sim \tau^{2H}$$

	AMZN	CAC
Close Prices		
Log Return		

HURST (1951)

Rescaled Range Analysis: capital sums of deviations from their means, rescaled by their standard deviation.

$$H = \frac{R(n)}{S(n)}$$

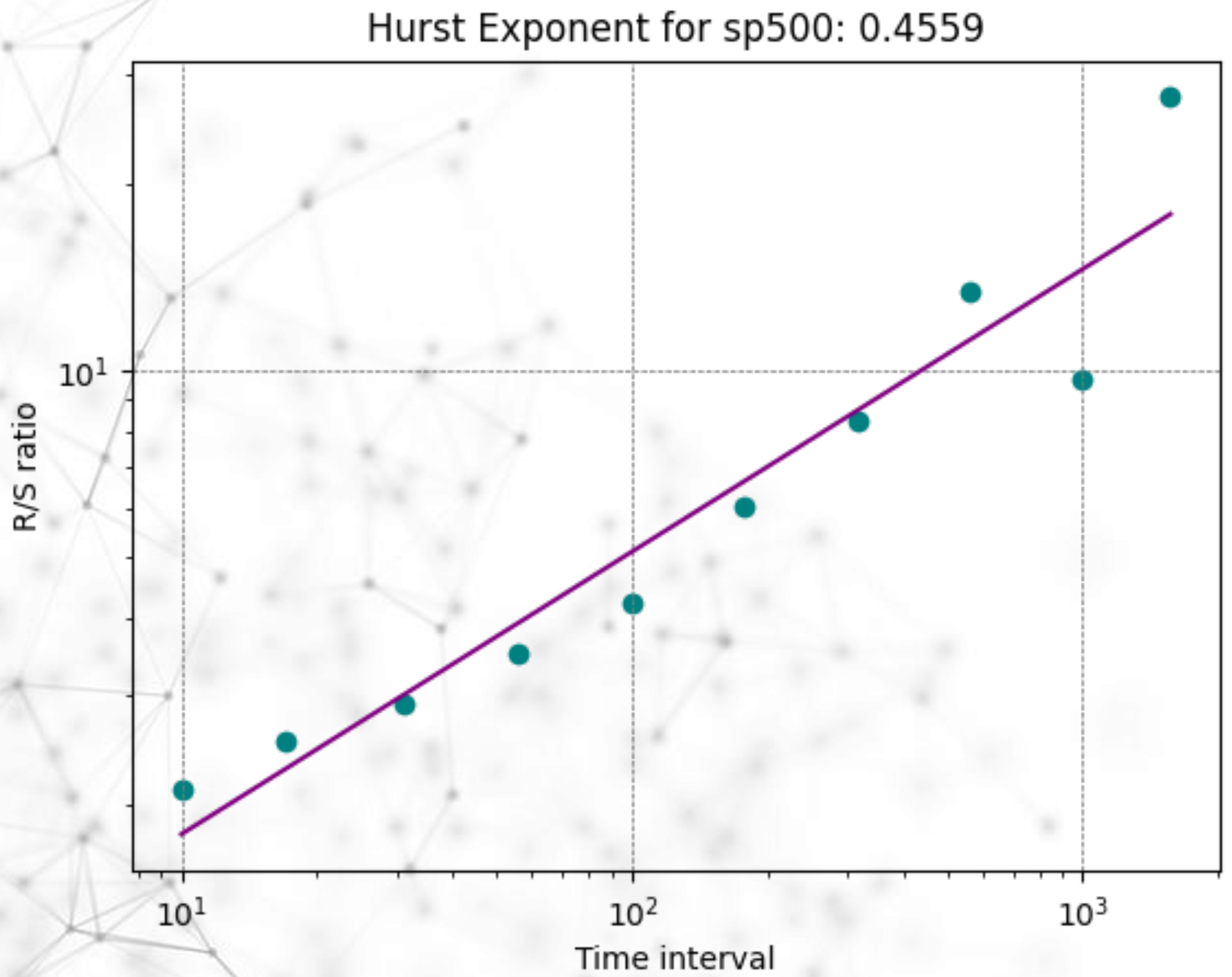
	AMZN	MSFT	IBM	SP500	NASDAQ	CAC
Hurst coeff of Close Price	0.025	0.053	-0.013	0.016	0.064	0.028
Hurst Coeff of Log Return	0.523	0.444	0.418	0.502	0.447	0.362
Hurst Coeff of Return	0.515	0.489	0.481	0.456	0.541	0.515

HURST (1951)

Rescaled Range Analysis: capital sums of deviations from their means, rescaled by their standard deviation.

H: *slope of*

$$approximated\ \frac{R(n)}{S(n)}$$



ANFIS MODEL

anfis-pytorch implementation by James Power

Turn it into a 3-label classifier

Adjust:
- Number of output variables
- Loss function
- Training: just BP, no Hybrid learning
- Add sigmoid MF

Predict next week's behaviour:
1. Bearish: -1
2. Flat: 0
3. Bullish: +1

Stationary features:
1. RSI
2. CPC
3. CCI
4. EMADN
5. Pct. Return

ANFIS MODEL

Target Variable

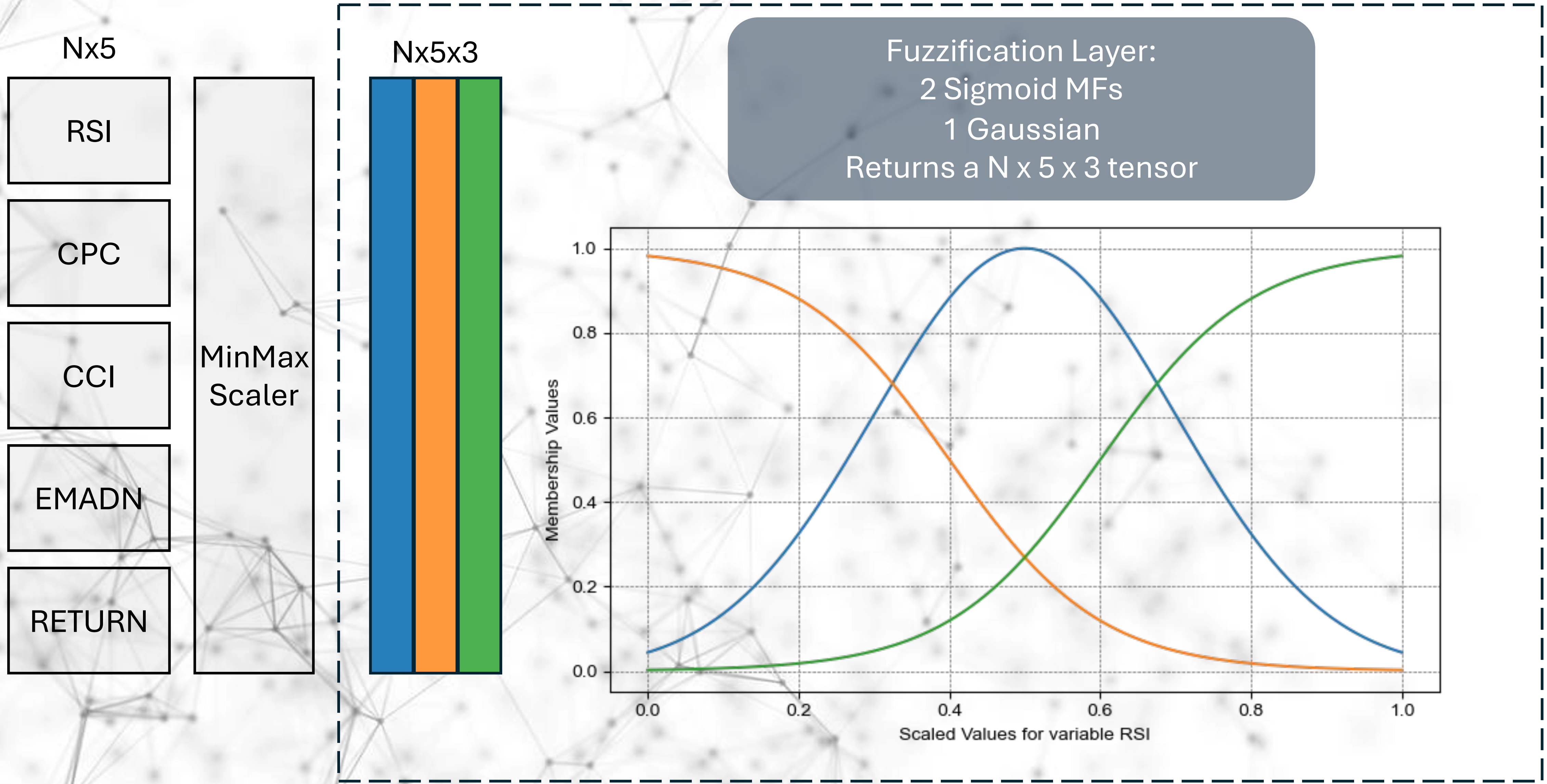
Created by calculating the percentage return between the next week's close price and the current week's

3 Possible actions:

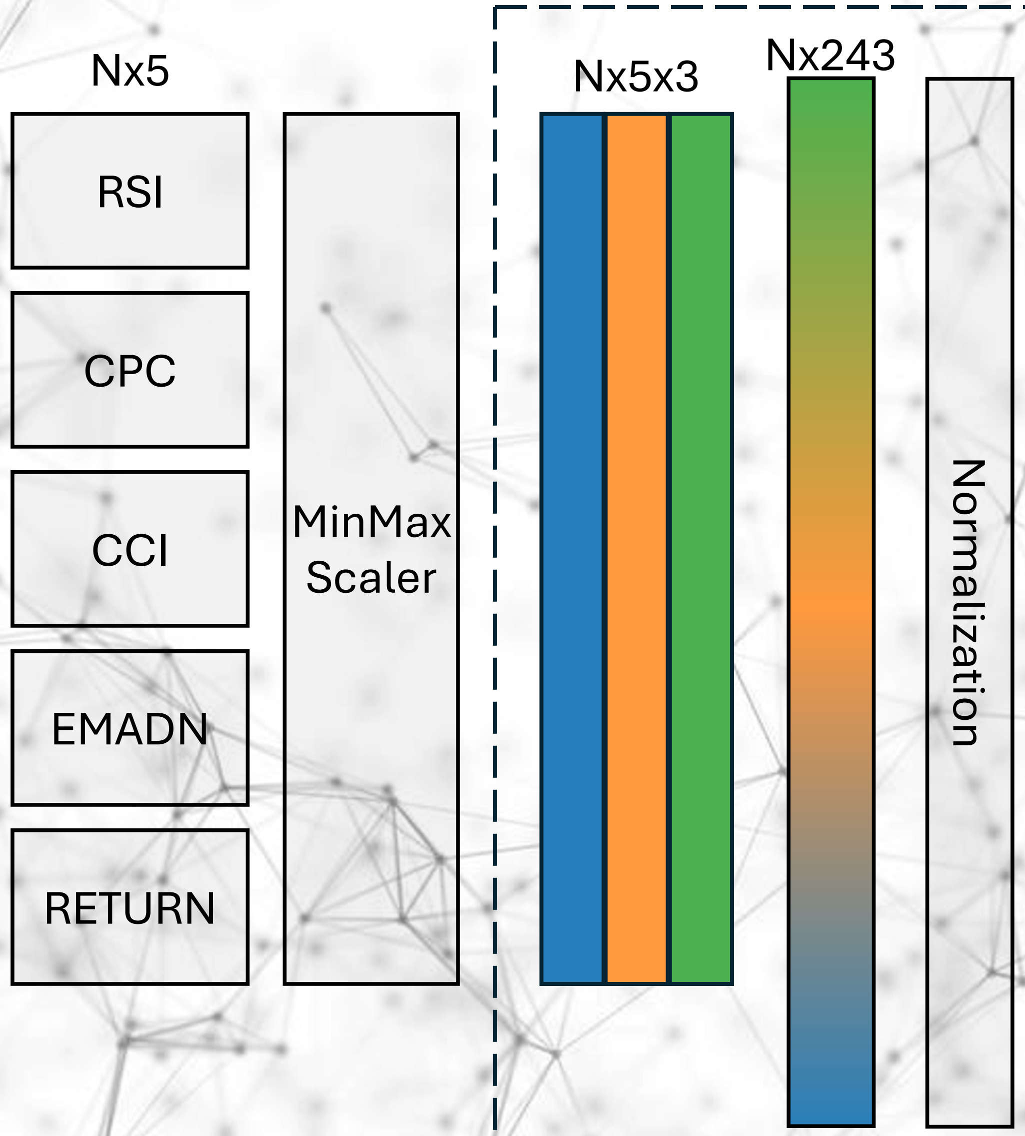
- 1 – Long
- -1 – Short
- 0 – Close position

- If percentage change is between +/- 0.5%, assign 0
- If bullish, i.e., percentage change in price for following week is positive, assign 1
- If bearish, i.e., percentage change in price for following week is negative, assign -1

ANFIS MODEL


Nx5
RSI
CPC
CCI
EMADN
RETURN
MinMax Scaler
Nx5x3
Fuzzification Layer:
2 Sigmoid MFs
1 Gaussian
Returns a N x 5 x 3 tensor
1.0
0.8
0.6
0.4
0.2
0.0
Membership Values
0.0
0.2
0.4
0.6
0.8
1.0
Scaled Values for variable RSI

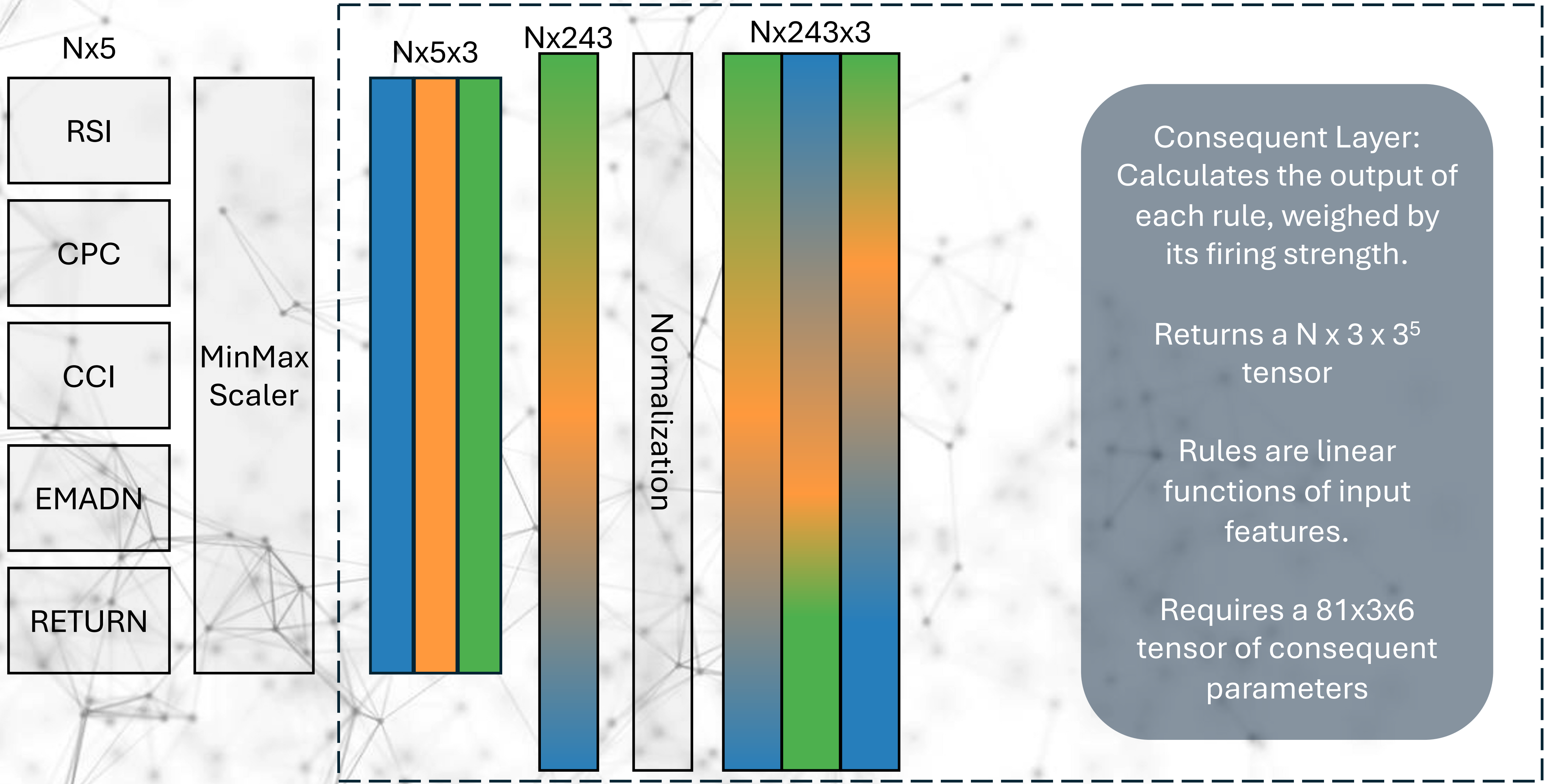
ANFIS MODEL



Antecedent Layer:
Returns N x 3^5 matrix with all possible pariwise products between membership values of different features.
These are ‘weights’ that serve as ‘firing strengths’ for the model’s rules

Normalization:
All weights normalized by their sum

ANFIS MODEL



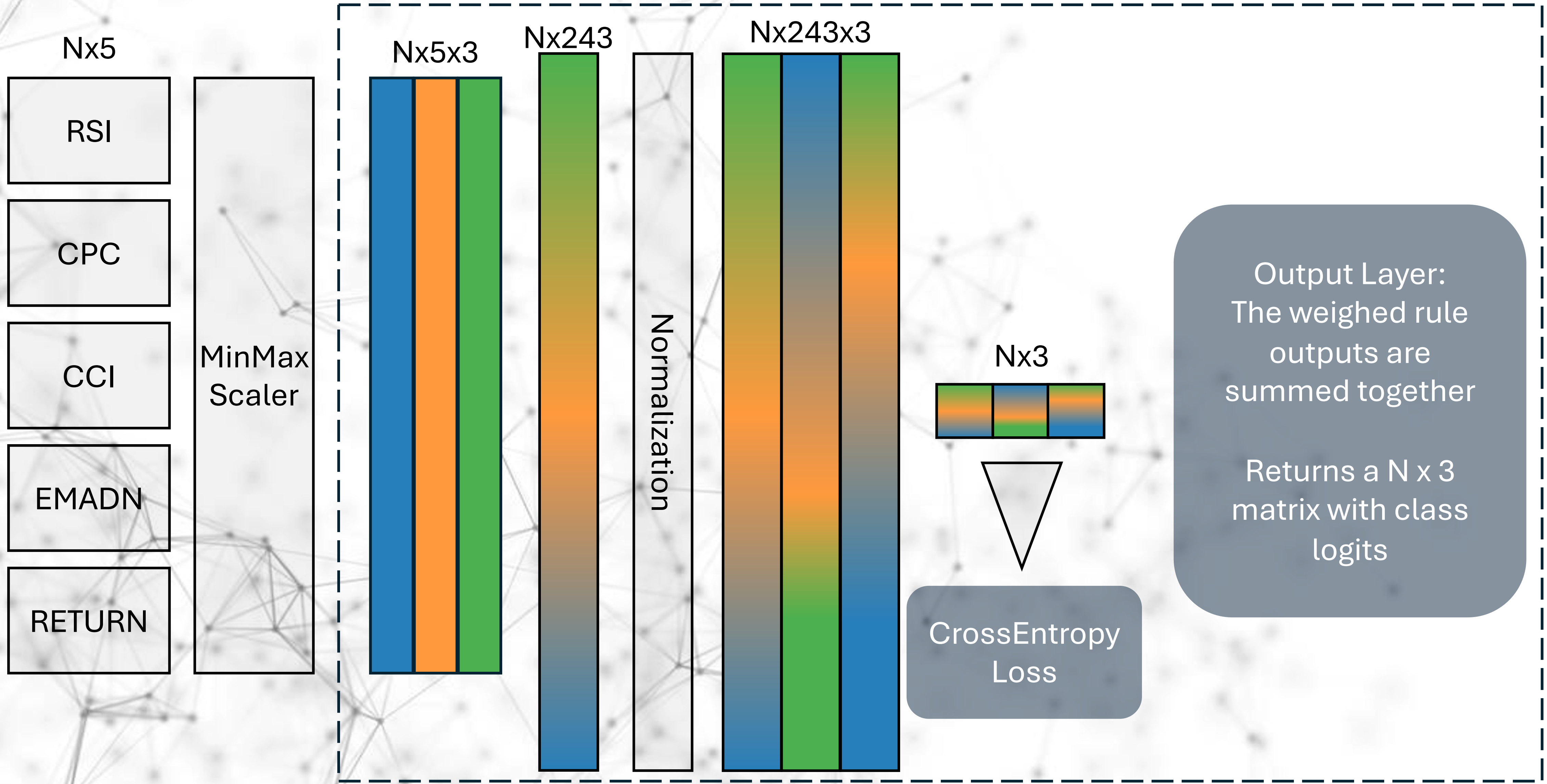
Consequent Layer: Calculates the output of each rule, weighed by its firing strength.

Returns a N x 3 x 3^5 tensor

Rules are linear functions of input features.

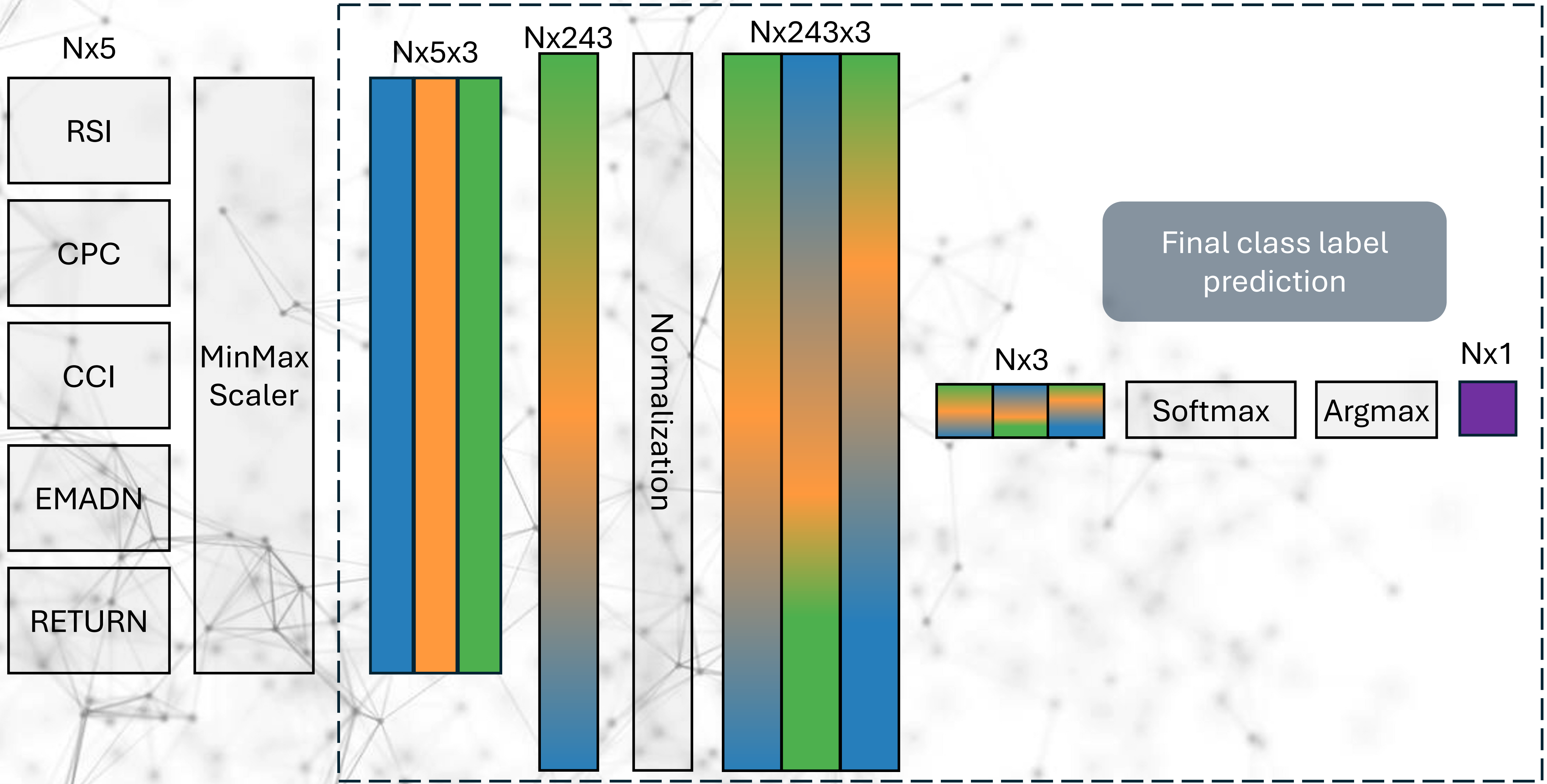
Requires a 81x3x6 tensor of consequent parameters

ANFIS MODEL



Nx5
RSI
CPC
CCI
EMADN
RETURN
MinMax Scaler
Nx5x3
Nx243
Normalization
Nx243x3
Nx3
CrossEntropy Loss
Output Layer:
The weighed rule outputs are summed together
Returns a N x 3 matrix with class logits

ANFIS MODEL

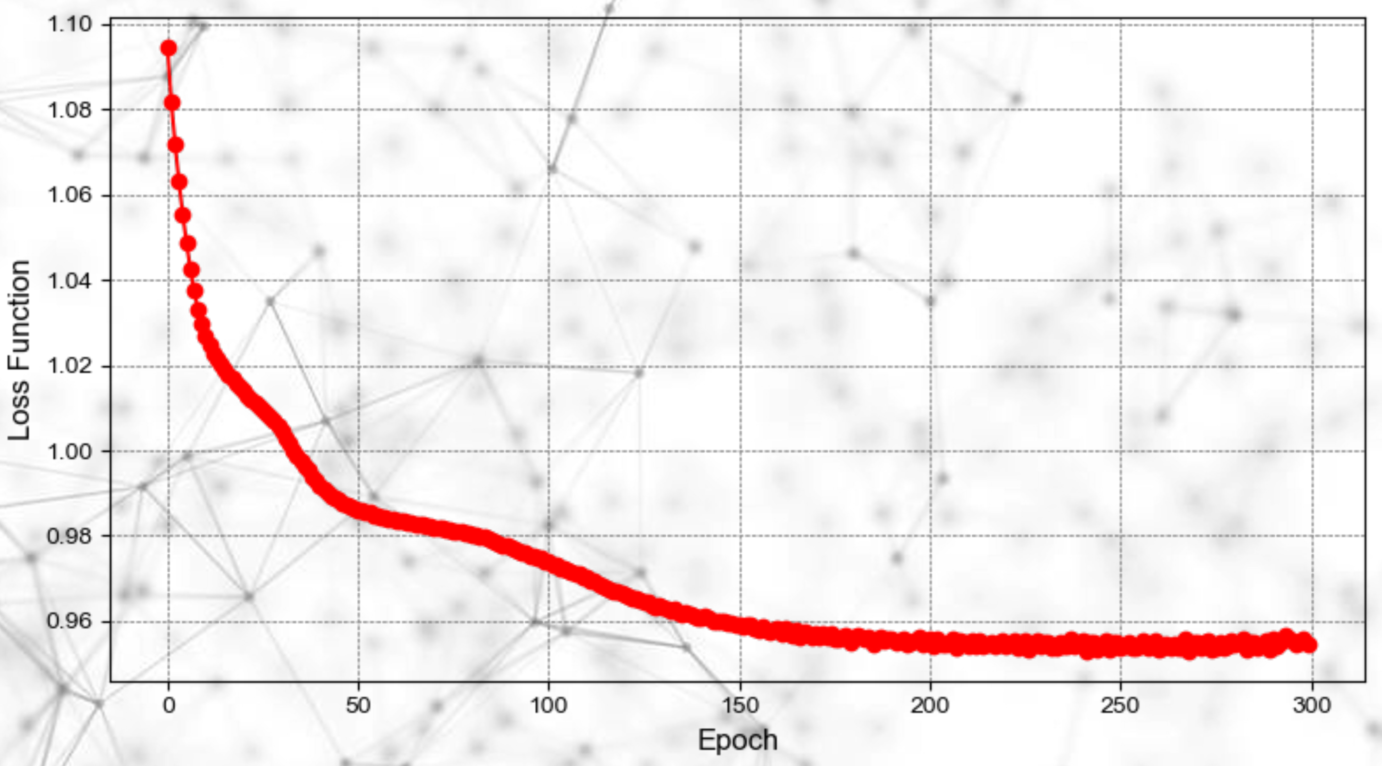


Nx5
RSI
CPC
CCI
EMADN
RETURN
MinMax Scaler
Nx5x3
Nx243
Normalization
Nx243x3
Final class label prediction
Nx3
Softmax
Argmax
Nx1

ANFIS MODEL

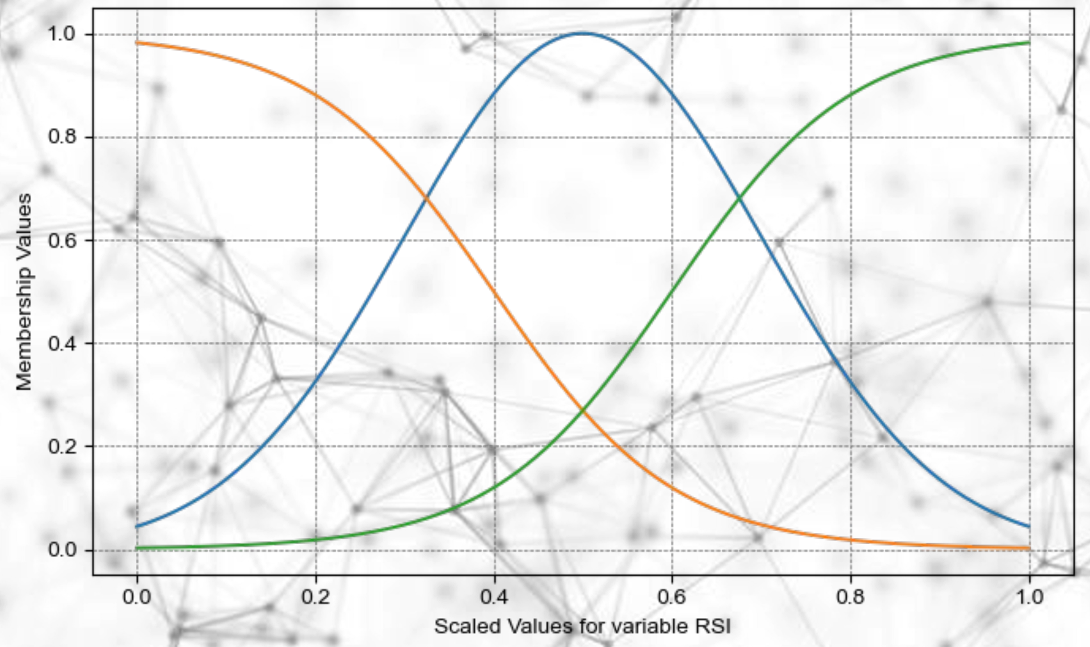
Training and validation on AMZN datset, 90/10 split:

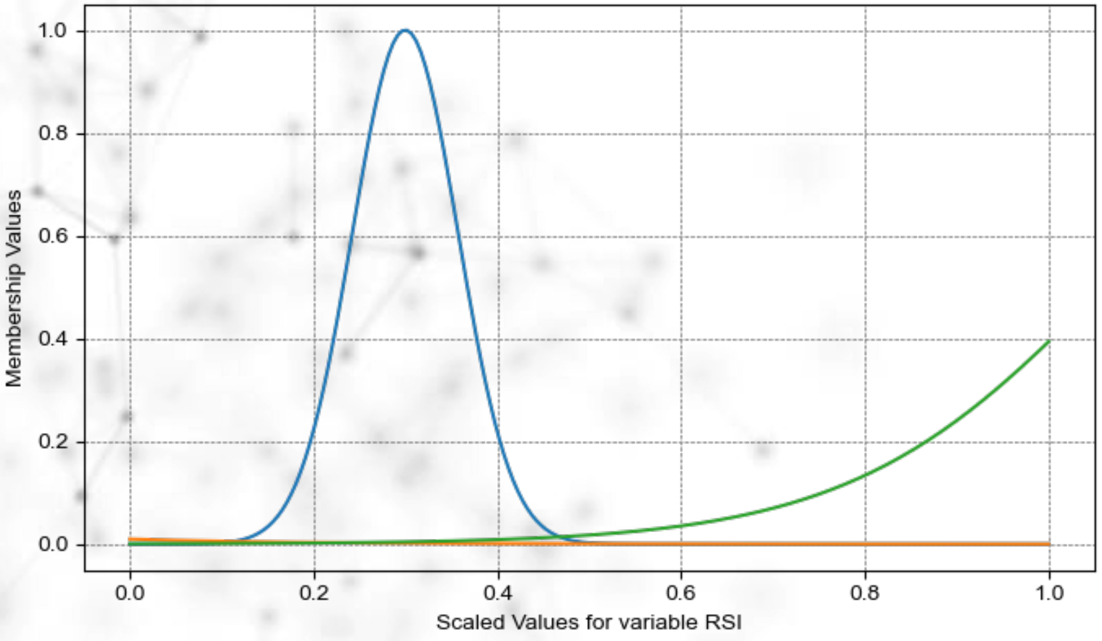
- Adam optimizer
- Gradient clipping
- LR decay



ANFIS MODEL

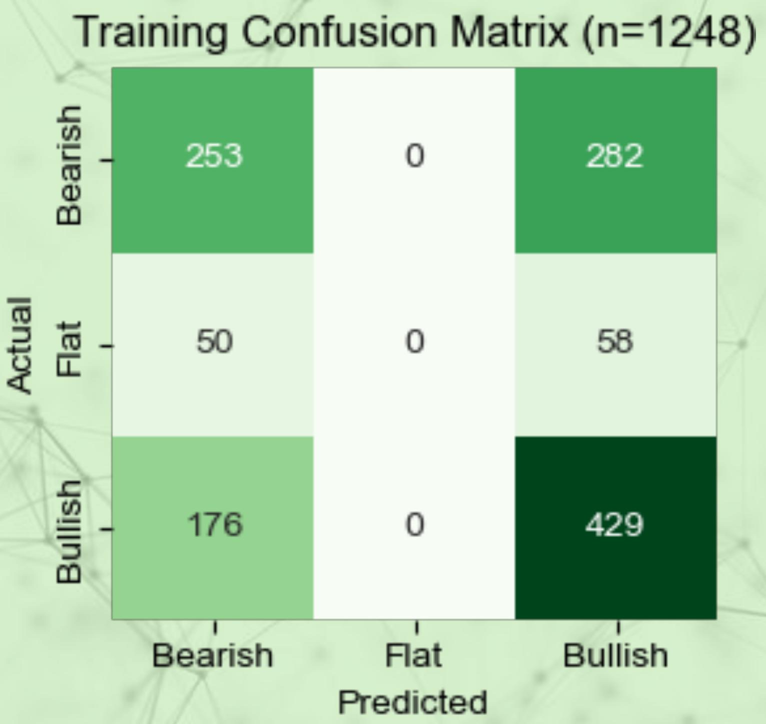
Training adjusts:
- Parameters of membership functions: 5x3x2=30
- Consequent layer parameters: 81x3x6=1,458

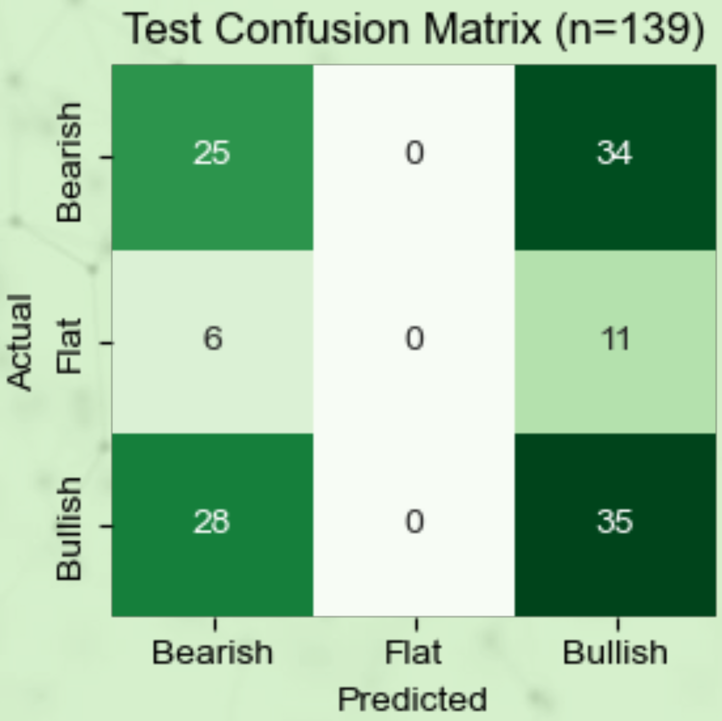




ANFIS MODEL

Results for AMZN:
Training: 55% accuracy, 0.52 F1 score
Validation: 43% accuracy, 0.40 F1 score





NEXT STEPS

- Use domain knowledge to tune feature parameters.
- Address issue with Flat Predictions.
- ANFIS Time Series Prediction.
- Calculate profit of the various Models.
- Finalize AI-Trading Model.

NEXT STEPS

Calculate profit of the various Models.

Profit Function:

$$\pi = \sum_{All\ Weeks} Output * (Close - Open)$$

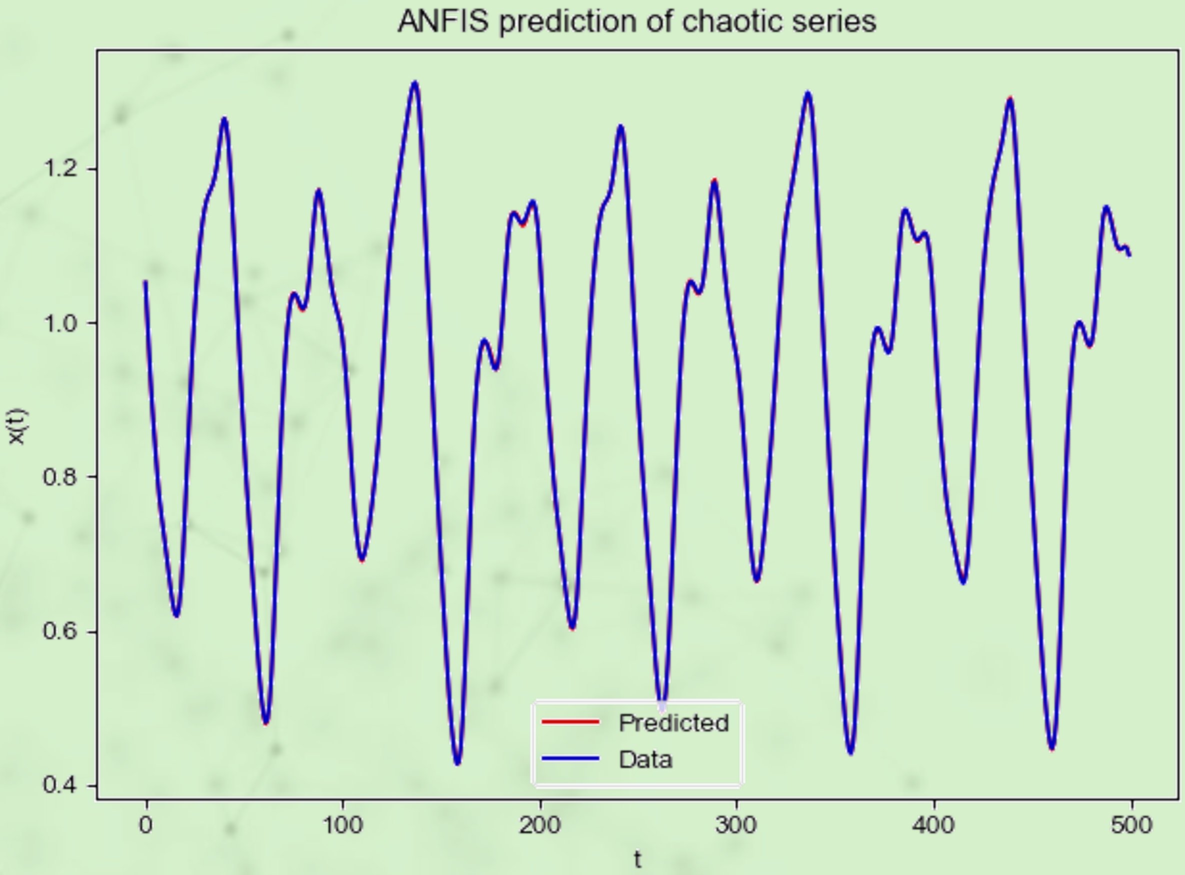
where $Output \in \{-1,0,1\}$ are predicted market behaviour labels, corresponding to trading actions

NEXT STEPS

ANFIS Time Series Prediction

In the original paper by Jang, ANFIS outperforms AR models in stationary time series prediction

Problem: Hurst Coefficient
The series used in the paper is definitely anti-persistent



THANK YOU FOR YOUR ATTENTION!

Gian Lorenzo Marchioni – 788811

David Pasquette – 789331

Elena Tomasella – 781321

LUISS

EUKLID

References

1. Carbone, A., et al. "Time-Dependent Hurst Exponent in Financial Time Series." *Physica A: Statistical Mechanics and Its Applications*, vol. 344, nos. 1-2, Dec. 2004, pp. 267-271, https://doi.org/10.1016/j.physa.2004.06.130.
2. Editor. "Detecting Trends and Mean Reversion with the Hurst Exponent." *Macrosynergy Research*, research.macrosynergy.com/detecting-trends-and-mean-reversion-with-the-hurst-exponent/.
3. "Introduction to ARIMA Models." *Duke.edu*, 2019, people.duke.edu/~rnau/411arim.htm.
4. Jang, J. S. "ANFIS: Adaptive-Network-Based Fuzzy Inference System." *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, 1993, pp. 665-685.
5. Jang, S. W., and Lee, S. H. "Stock Forecasting using Fuzzy Neural Networks, Technical Indicators, and Foreign Exchange Rates." *International Journal*, vol. 9, no. 3, 2020.
6. Power, J. F. “ANFIS in pyTorch”. Maynooth University, 5 Jun 2019, https://github.com/jfpower/anfis-pytorch
7. Marton, Bence, and Cakir, Hasan. "Usage of the Hurst Exponent for Short Term Trading Strategies." 1 Dec. 2022.
8. "Understanding Indicators in Technical Analysis." Fidelity, 2020. https://www.fidelity.com/bin-public/060_www_fidelity_com/documents/learning-center/Understanding-Indicators-TA.pdf
9. Yadav, Anita, et al. "Optimizing LSTM for Time Series Prediction in Indian Stock Market." *Procedia Computer Science*, vol. 167, 2020, pp. 2091-2100, https://doi.org/10.1016/j.procs.2020.03.257.
10. Zhang, G. Peter. "Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model." *Neurocomputing*, vol. 50, Jan. 2003, pp. 159-175, www.sciencedirect.com/science/article/pii/S0925231201007020, https://doi.org/10.1016/s0925-2312(01)00702-0.