VILNIUS UNIVERSITY
FACULTY OF MATHEMATICS AND INFORMATICS
INSTITUTE OF COMPUTER SCIENCE
DEPARTMENT OF COMPUTATIONAL AND DATA MODELING

Signal Analysis and Processing

Moving average algorithm analysis

Done by:
Kazimieras Vitkus

Supervisor:
dr. Tadas Meskauskas

Vilnius
2024

Contents

1 Introduction

The task of this project is to implement, analyse and compare moving average algorithms. Two algorithms are in the scope of the project: Weighted Moving average (WMA) and Exponential Moving Average (EMA). The algorithms will be implemented in Python programming language, using NumPy, Pandas, Matplotlib libraries. The programming will be done in Jupyter Notebook environment. The algorithms will be tested on the same dataset.

2 Definition of algorithms and signal

2.1 Moving Average algorithm

A moving average algorithm is a statistical technique used to analyze sequential data points by creating a series of averages of different subsets of the full dataset. Typically applied in time-series analysis, finance, and signal processing, it smooths out fluctuations to reveal underlying trends or patterns. The algorithm involves calculating the mean of a specified number of data points within a sliding window, which moves along the dataset, updating the average at each step. This process effectively filters out short-term fluctuations, making it easier to identify longer-term trends or patterns within the data.

Given that we have a digital signal $f_0, f_1, ..., f_N$ and chosen the window size K - positive integer. By applying the moving average algorithm to f_i we get the sum of averages in window of size K to both side. The formula for moving average is:

$$g_i = \frac{1}{2K+1} \sum_{j=-K}^{K} f_{i+j}, \qquad i = K, K+1, ..., N-K \tag{2.1}$$

To be noted, the first value of signal that has been averaged value is f_K and the last value is f_{N-K}. The moving average algorithm is used to smooth out short-term fluctuations in the data, however during the process the length of the signal is reduced.

2.2 Weighted Moving Average algorithm

In the subsection 2.1 moving average algorithm was introduced. The moving average algorithm is simple, due to the fact that it averages all the values in the window equally as shown in the formula 2.1. However, in some cases it is more beneficial to give more weight to the values that are closer to the current value. This is where the weighted moving average algorithm comes in. The weighted moving average algorithm is a variation of the moving average algorithm, where the values in the window are given different weights. The formula for weighted moving average is:

$$g_i = \frac{1}{2K+1} \sum_{j=-K}^{K} w_j f_{i+j}, \qquad i = K, K+1, ..., N-K, \qquad w_j \sum_{j=-K}^{K} = 1 \tag{2.2}$$

The weighted moving average algorithm weights can be defined in various ways. The most common way is to use the Gaussian function. The Gaussian function is a bell-shaped curve that is symmetric around the mean. The Gaussian function is defined as:

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \tag{2.3}$$

Lets say that f_{i-K} and f_{i+K} should be L times less important than the center value f_i. In this case $L > 1$ is a positive integer, a parameter to weighted moving average algorithm. The range of weights distribution could be discretised in range: $-x_k \leq x \leq x_k$, then it can be calculated where x_k, is less than center value x_0 in Gaussian distribution formula 2.3:

$$\frac{p(0)}{p(x_k)} = L, \qquad x_k = \sqrt{-2 \ln L} \tag{2.4}$$

To comply with the rule that weights $\sum_{j=-K}^{K} w_j = 1$ the weights should be normalized. The normalized weights are defined as:

$$w_j = \frac{p(j)}{\sum_{j=-K}^{K} p(j)}, \qquad j = -K, -K+1, ..., K \tag{2.5}$$

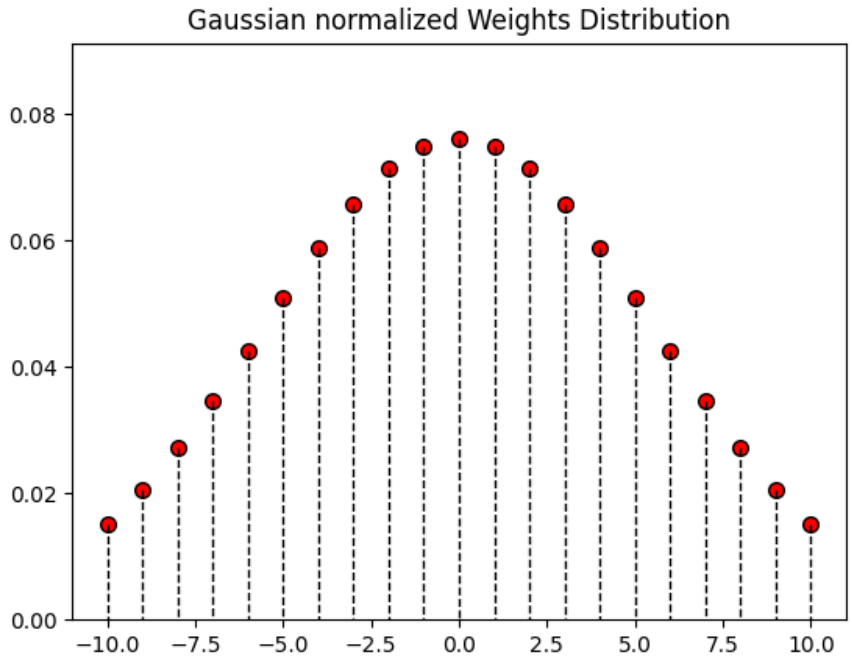

Figure 1. Distribution of normalized weights where L = 10 and K = 30

2.3 Exponential Moving Average algorithm

The exponential moving average algorithm is another variation of the moving average algorithm. This method is similar to the weighted moving average algorithm 2.2, but have two main diffrences:

1. Instead of parameters K and L the exponential moving average algorithm has only one parameter α.

2. When new signal value is appended, it is enough to know the previous average value to calculate the new average value. For refrence, other averaging algorithms need to know all the values in the window.

To average the signal with exponential moving average algorithm, the formula is used:

$$g_n = (1 - \alpha)g_{n-1} + \alpha f_n, \qquad n = 1, 2, \dots, \qquad g_0 = f_0 \tag{2.6}$$

2.4 Signal for the algorithm analysis

For the task of this project, the financial market signal was chosen. The daily values of the S&P 500 index[1] at market close from 2014 to 2024 were used. The signal was chosen because it has high volatility and represents the financial market well.

Figure 2. S&P 500 index signal from 2014 to 2024

As can be seen from figure 2, the value of the index have increased over 2.5 times in the period of 10 years. The signal has high volatility, which is good for the analysis of moving average algorithms.

3 Implementation

All of the algorithms were implemented in Python programming language, using libraries such as NumPy, Pandas, Matplotlib. The programming was done in Jupyter Notebook environment.

Appendix 1 contains the full Jupyter Notebook file with the implementation and analysis of the algorithms.

3.1 WMA. Weights

```
def gaussian_distribution(x):
    return 1/(np.sqrt(2*np.pi)) * np.exp(-((x**2)/2))


def get_weights(L,K):

#boundary value for weights
    x_k = np.sqrt(2*np.log(L))

#creating X axis for gaussian weight distribution
    K_j = np.linspace(-K,K,2*K+1)


#distributing weights along X axis
    x_j = [x_k * j/K for j in K_j]

# applying gaussian distribution to weights
    p_j = [gaussian_distribution(x) for x in x_j]

#normalization of weights
    w_j = [p/sum(p_j) for p in p_j]
    return K_j,w_j,p_j
```

3.2 WMA. Algorithm

```
def apply_weights(data, weights):
        weighted_data = 0
        for i in range(len(data)):
            weighted_data += data[i] * weights[i]
        return weighted_data

def weighted_moving_average(data, L, K):

    K_j,weights,p_j= get_weights(L,K)
    weighted_data = np.zeros(len(data))

    for i in range(K, len(data) - K):
        window_data = data[i - K : i + K + 1]
        weighted_data[i] = apply_weights(window_data, weights)

   weighted_data = np.where(weighted_data == 0, np.nan, weighted_data)

    return weighted_data
```

3.3 EMA. Algorithm

```
def exponential_moving_average(data, alpha):
    averaged_data = np.zeros(len(data))
    averaged_data[0] = data[0]

    for i in range(1, len(data)):
      averaged_data[i] = alpha * data[i] + (1 - alpha) * averaged_data[i -
1]

    return averaged_data
```

4 Analysis

4.1 Averaging signal with WMA

The weighted moving average algorithm has two parameters that can be adjusted

1. **K** - the size of the window;
2. **L** - the parameter that defines the importance of the values in the window.

By adjusting these parameters, the algorithm tends to smooth out the signal more or less. However, by over tuning the parameters, the algorithm can lose the important information in the signal, i.e. the algorithm can over smooth the signal.

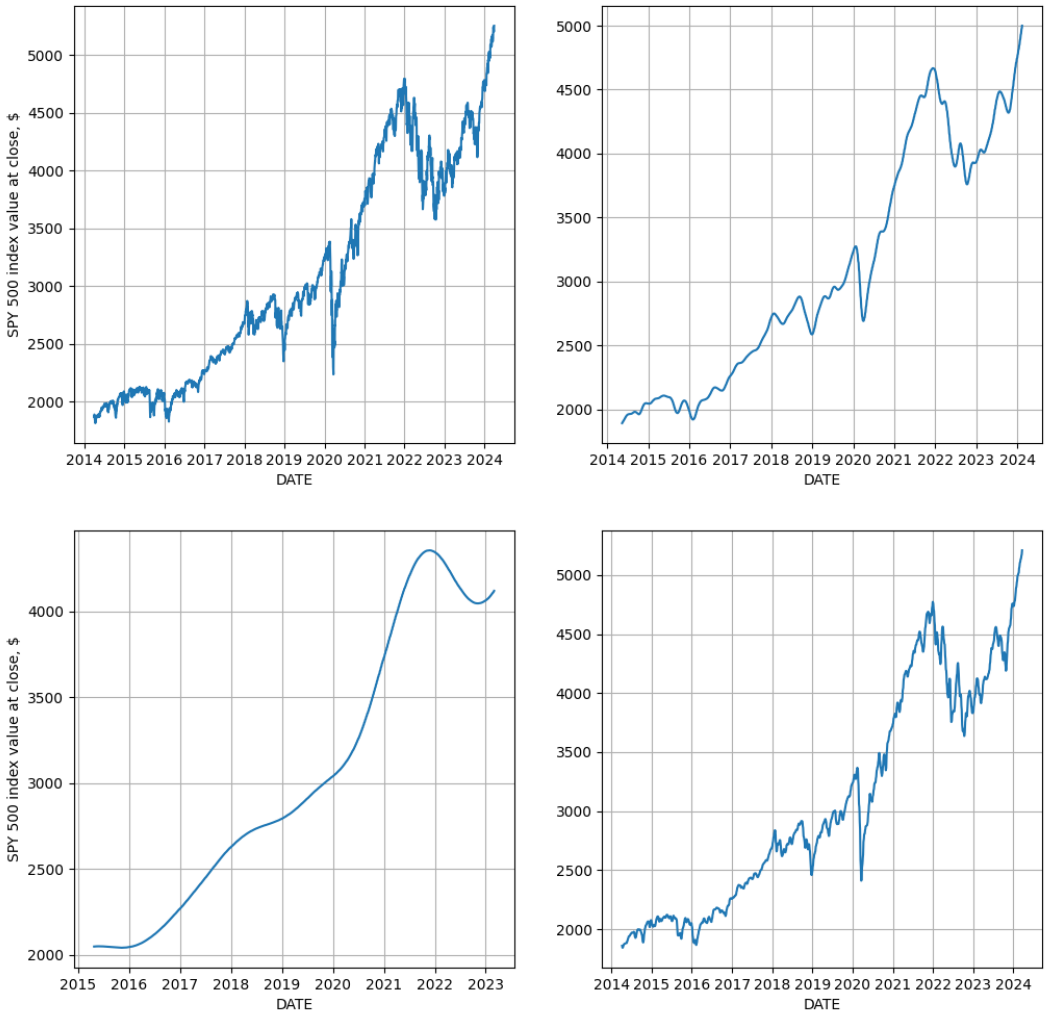

Figure 3. On top left chart original signal is shown, on top right chart signal is averaged with WMA with K = 30 and L = 10, on bottom left chart signal is averaged with WMA with K = 270 and L = 30, on bottom right chart signal is averaged with WMA with K = 5 and L = 5.

As can be seen from the figure 3, the parameters adjust the smoothness of the signal. The larger the K the smoother the signal gets, also the window of values get shorter,due to algorithms design that only allows to compute values that are in range of $g = (f_{0+K}; f_{N-K})$, see the top left and the bottom left charts in figure 3. The larger the L the more the values in the window are weighted towards the center value.

For this particular case study of SP500 signal, the logic parameters for WMA should correlate with the cycles of weeks, months, quarters and years. For the case of WMA, parameter K represents the days that are in window, for example if we set $K = 30$ the window will be 60 days, which is approximately one month before and one month after to the date of intrest.

4.2 Qualitative analysis of WMA

The main concerns with moving average algorithms is that they tend to lag behind the signal, can be too smooth or retain too much noise. It all depends on the parameters that are chosen for the algorithm. The weighted moving average algorithm is more flexible, because it allows to adjust the weights of the values in the window.

There some methods to evaluate the quality of the algorithm:

1. Visual inspection of the signal;
2. Comparison of the averaged signal with the original signal;
3. Evaluating the correlation between the original signal and the averaged signal;
4. Evaluating the gradient variance and covariance of the original signal and the averaged signal.

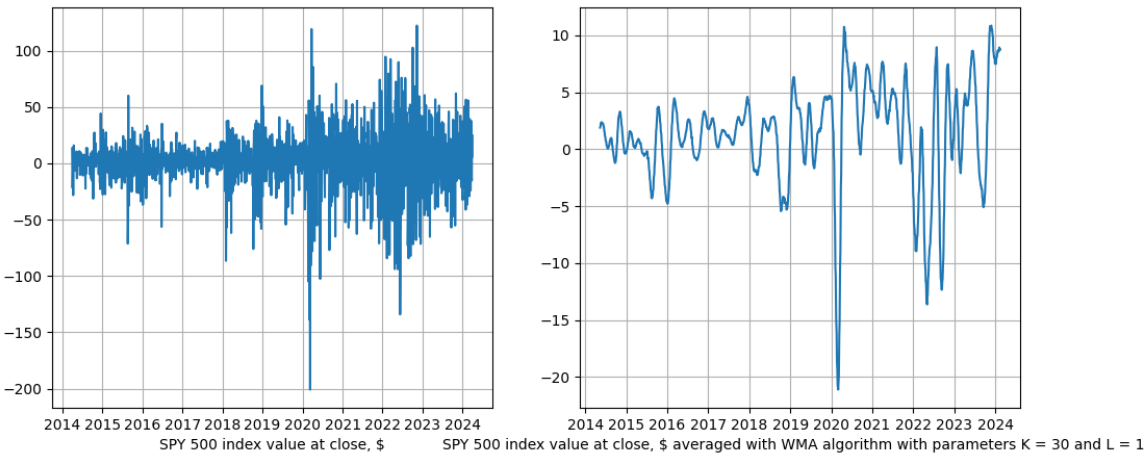

Figure 4. In top chart the variance of original signal is shown with variance of = 566.397, in the bottom chart the variance of signal averaged with WMA with K = 30 and L = 10 is shown with variance of = 19.677.

As depicted in figure 4, the variance of the signal is reduced by the weighted moving average algorithm. The gradient of the signal variance, can help to evaluate if the smoothness of the signal fits the requirements. The gradient of the variance can be calculated as:

$$\Delta f = \frac{f_{n+1} - f_n}{f_n} \tag{4.1}$$

To evaluate if the averaged signal retains the trends, the correlation between the original signal and the averaged signal can be calculated. The correlation can be calculated as:

$$\rho = \frac{cov(f, g)}{\sqrt{var(f)var(g)}} \tag{4.2}$$

	K=7	**K=30**	**K=90**	**K=180**	**K=360**
L=1	0.951	0.814	0.583	0.391	0.209
L=5	0.951	0.815	0.583	0.395	0.221
L=10	0.951	0.815	0.583	0.396	0.224
L=20	0.951	0.815	0.583	0.397	0.227
L=30	0.951	0.816	0.583	0.397	0.229
L=50	0.951	0.816	0.584	0.398	0.231

Table 1. Matrix of WMA parameters K and L impact on correlation

As could be seen from the table 1, the correlation between the original signal and the averaged signal is dependant on the parameter K. The parameter L has negligible impact on the correlation.

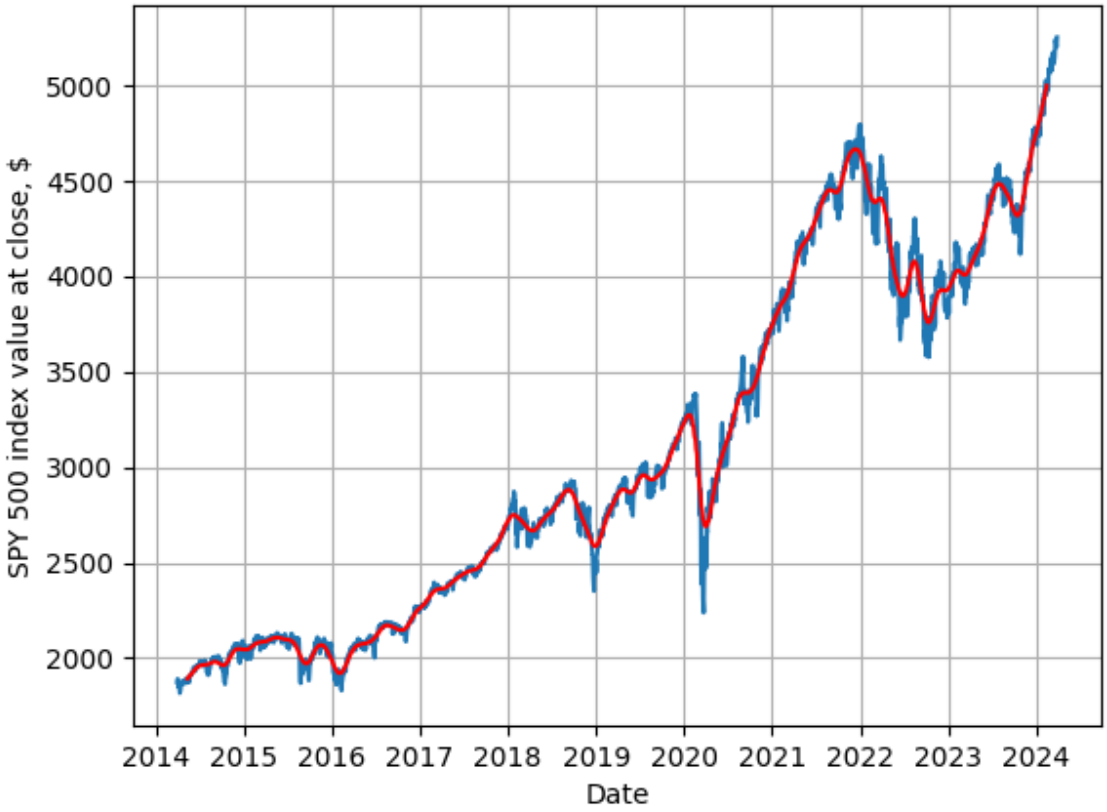

Figure 5. S&P 500 signal from 2014 to 2024 overlayed with signal averaged with WMA with K = 30 and L = 10

4.3 Averaging signal with EMA

EMA has only one parameter α, which is the weight of the new value in the signal. However, it retains the same issues as WMA - if the weight is not chosen correctly, the algorithm can lag behind the signal, be too smooth or retain too much noise.

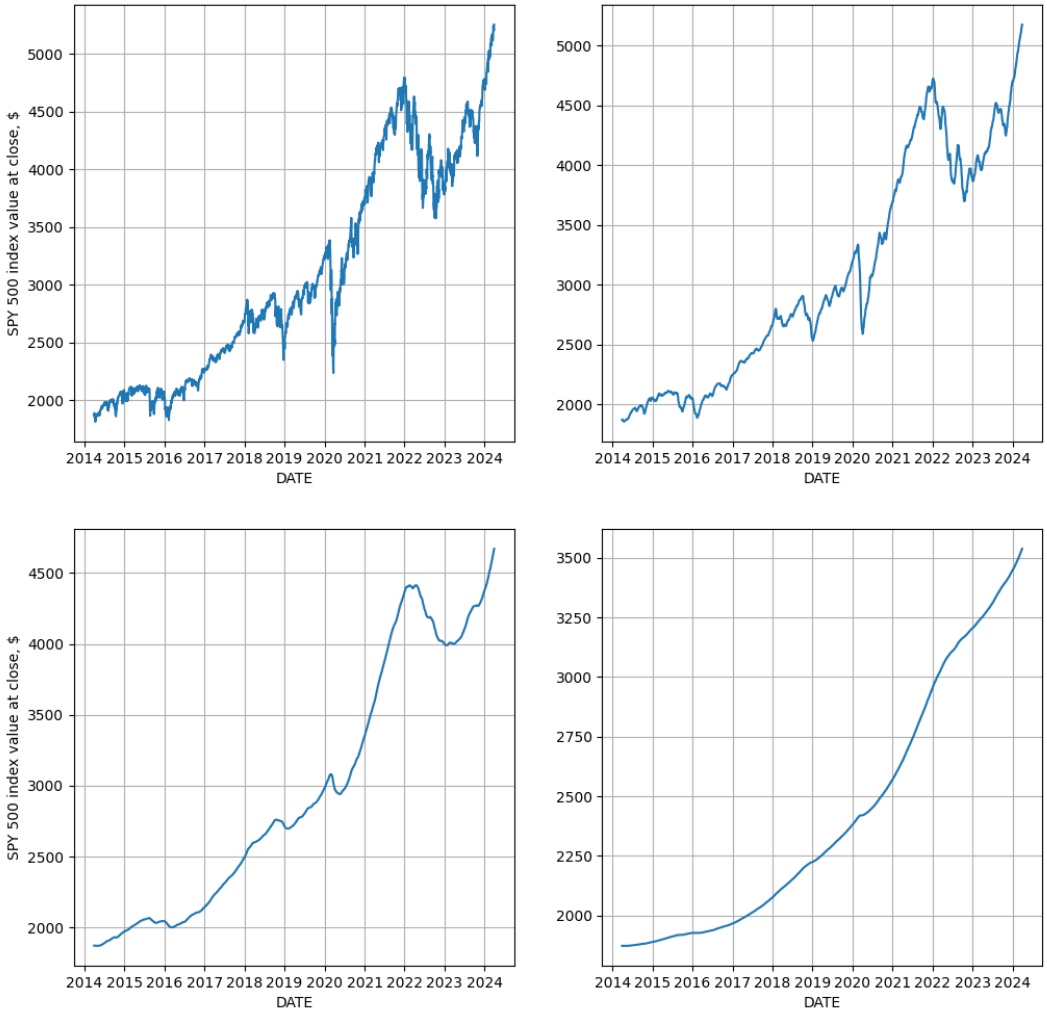

Figure 6. On top left chart original signal is shown, on top right chart EMA is applied with $\alpha = 0.1$, on bottom left chart EMA is applied with $\alpha = 0.01$, on bottom right chart EMA is applied with $\alpha = 0.001$.

As can be seen from the figure 6, the smaller the α the smoother the signal gets and lag occurs.

Although the EMA algorithm has only one parameter and is more difficult to fine tune, but it has an advantage over the WMA algorithm - it does not require all the values in the window to calculate the new average value. And even next value can be predicted if previous value is known.

4.4 Qualitative analysis of EMA

The EMA algortithm tends to be more tough for fine tuning, but the same evaluation techniques can be adapted for EMA as for WMA. The correlation between the original signal and the averaged signal can be calculated as:

Alpha	Covariance	Correlation
0.1	51.67	0.997
0.08	40.53	0.996
0.06	29.38	0.995
0.04	18.43	0.993
0.02	8.29	0.988
0.01	3.72	0.979
0.005	1.50	0.967
0.001	0.22	0.944

Table 2. Covariance and correlation between original signal and signal averaged with EMA with different α values

As can be seen from the table 2, the EMA algorithm tends to retain the trend of the signal better than the WMA algorithm. However, the signal gets smooth in a very short increments of the parameter change.

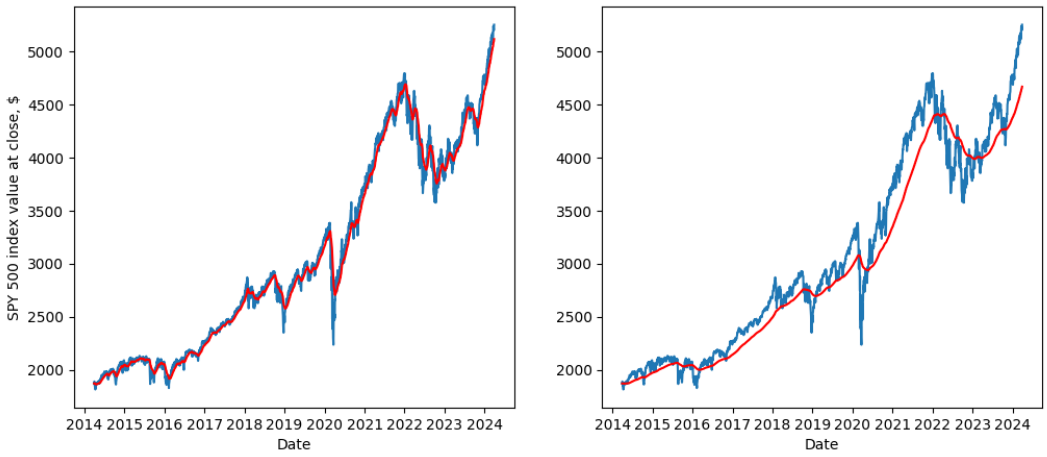

Figure 7. S&P 500 signal from 2014 to 2024 overlayed with signal averaged with EMA with $\alpha = 0.06$ on the left side and with $\alpha = 0.01$ on the right side.

As can be seen from the figure 7, EMA algorithm also tends to lag behind the signal when the parameter α is too small.

4.5 Execution times

One of the important aspects of the algorithms - execution time savings.

N	WMA	EMA
100	0.0016	0.0004
1000	0.0271	0.0014
10000	0.1709	0.0052
50000	0.6697	0.0295
100000	1.4398	0.0528

Table 3. Execution times for WMA with parameters L = 5 and K = 20 and EMA with parameter $\alpha = 0.05$ algorithms in seconds

As can be seen from the table 3, EMA is way faster than WMA. The execution time of the EMA algorithm is not dependant on the size of the signal, because it only requires the previous value to calculate the new average value. The execution time of the WMA algorithm is dependant on the size of the parameter K - size of the window - the bigger the K - more operations to compute in a single iteration. This could be deduced from the formula 2.2.

K	Execution time
5	0.0403
50	0.2524
100	0.4755
500	2.3591
1000	3.3062

Table 4. WMA execution time dependance on K parameter. In this example signal with length N = 10000 and parameter L = 5

As could be seen in table 4, execution time of WMA linearly correlates with the size of parameter K.

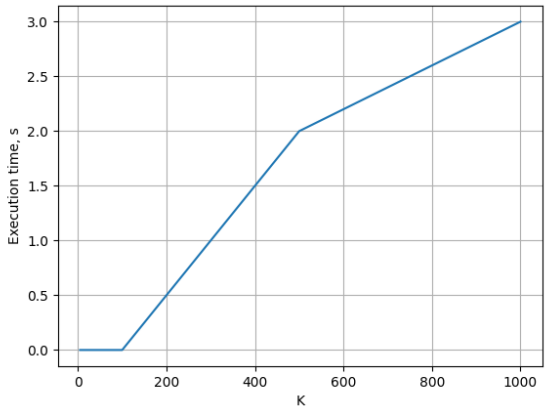

Figure 8. WMA execution time dependance on K parameter

4.6 Examples with other signals

To demonstrate the moving average algorithms versatilty, examples with other signals that are based on other domains will be shown.

- Signal 1 - ECG signal[2] - the signal is just a demonstrattive representation of the ECG signal, i.e. this is not real,but generated data. Due to the length of the signal, only a part of the signal

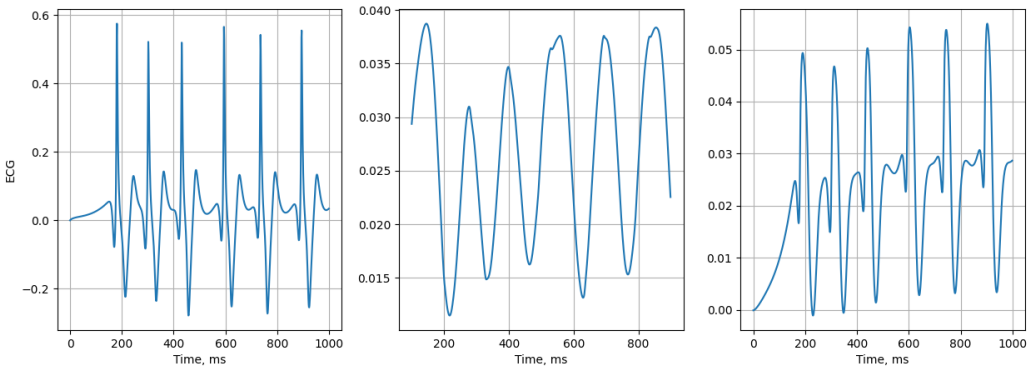

Figure 9. On most left chart the original ECG signal is presented, in the middle chart the ECG signal is averaged with WMA parameters L = 30 K = 150, on the most right, an ECG signal is averaged with EMA $\alpha = 0.01$

is shown in the figure 9. The ECG signal is a good example of the signal that has a lot of noise, and the moving average algorithms can be used to smooth out the signal.

- Signal 2 - Personal saving rates in the USA[3] - the signal represents the percentage of persons income deduced to savings.

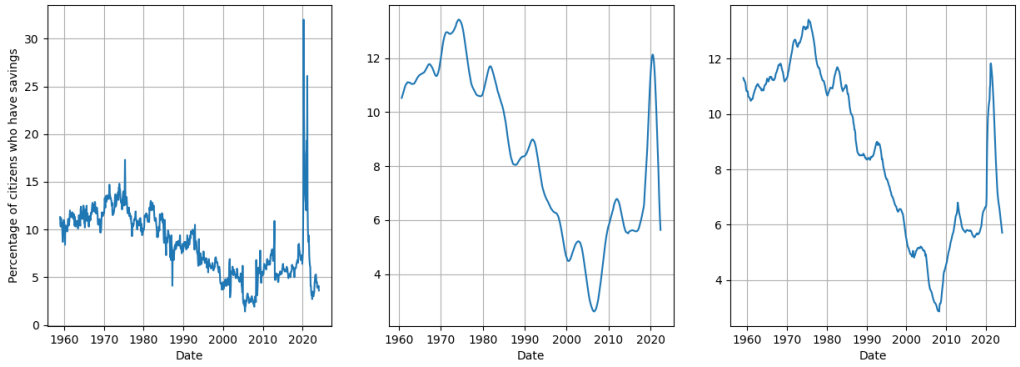

Figure 10. On most left chart the original signal of personal saving rates in the USA is presented, in the middle chart the signal is averaged with WMA parameters L = 5 K = 20, on the most right, the signal is averaged with EMA $\alpha = 0.05$

As could be seen from the figure 10, the signal when averaged manages retain the trend, however, it cuts short the 2020 peak.

- Signal 3 - weekly petrol prices in Italy[4]

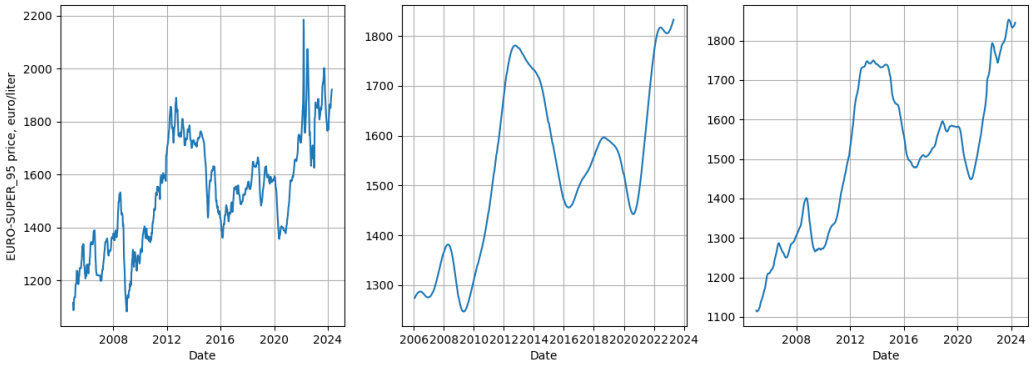

Figure 11. On most left chart the original signal of weekly petrol prices in Italy is presented, in the middle chart the signal is averaged with WMA parameters L = 24 K = 52, on the most right, the signal is averaged with EMA $\alpha = 0.03$

- Singal 4 - random generated signal.

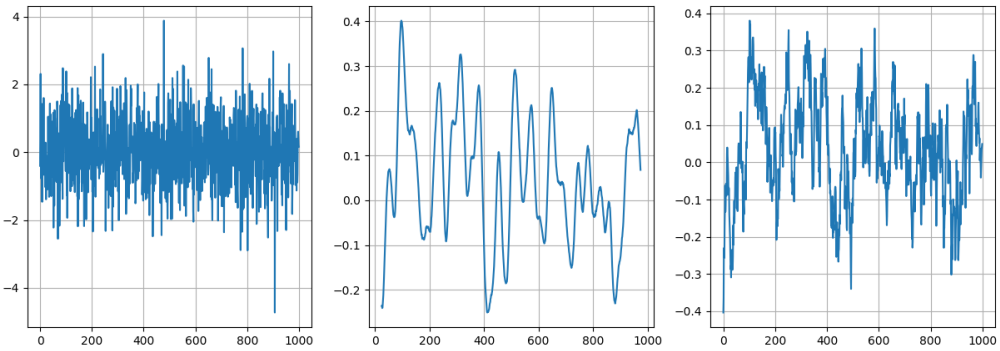

Figure 12. On most left chart the original random signal is presented, in the middle chart the signal is averaged with WMA parameters L = 20 K = 25, on the most right, the signal is averaged with EMA $\alpha = 0.05$

5 Conclusion

In this task two algorithms were implemented, WMA and EMA, they were analysed, qualitative parameters such as correlation and variation were measured and compared in between.

The WMA is more flexible, due to the fact that is has two parameters for optimizations, however, some values are lost in analysis, because the averaging is done in range of windows.

The EMA is more robust, because it does not require all the values in the window to calculate the new average value. The EMA can be used to predict the next upcoming value, if the previous value is known.

One huge advantage EMA has over WMA - execution time. If the signal is large, the EMA algorithm is way faster than the WMA algorithm. However, the EMA is more difficult to fine tune, because it has only one parameter. To increase the usability of the EMA, variating parameter α could be used, but this was not in the scope of the project.

References

[1] S&P Dow Jones Indices LLC, S&P 500 [SP500], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/SP500,

[2] ECG signal demo; https://www.kaggle.com/datasets/ahmadsaeed1007/ecg-signal?resource=download

[3] U.S. Bureau of Economic Analysis, Personal Saving Rate [PSAVERT], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/PSAVERT

[4] Petrol prices in Italy; https://dgsaie.mise.gov.it/open-data