Monte Carlo Simulation and Geometric Brownian Motion Modeling for Stock Price Prediction in the US and European Markets

Isha Jagani

December 21, 2023

I. Introduction

The financial market's complexity and uncertainty make it an ideal candidate for Monte Carlo Simulations. This project explores the predictive power of Monte Carlo Simulation and Geometric Brownian Motion (GBM) modeling in forecasting stock prices in the US and European markets. The project includes Python programming and financial data from Yahoo Finance for S&P 500 (representative of the US market) and STOXX 50 (representative of the European market). The project uses simulations and visualizations to analyze true prices and Monte Carlo-simulated prices for both markets. The parameters and the models for the Monte Carlo Simulation are obbatined from the research paper "Monte Carlo Simulation Prediction of Stock Prices" by J. N. P. Xiang, S. R. Velu and S. Zygiaris.

II. Model Used

In this project we make use of two main models, the Monte Carlo Simulation and Geometric Brownian Motion (GBM) modeling. The models and parameters are obtained from the paper "Monte Carlo Simulation Prediction of Stock Prices" by J. N. P. Xiang, S. R. Velu and S. Zygiaris. According to the paper the GBM model, is a continuous-time stochastic process, in which the logarithmic of the randomly varying quantity follows a Brownian motion with drift. GBM is a continuous-time stochastic process that satisfies the Stochastic Differential Equation (SDE):

$$dSt = \mu Stdt + \mu StdBt \text{ (1)}$$

According to the paper in this equation, St is a stochastic process, and Bt is a Brownian motion with the following properties:

- B0 = 0
- B possesses both stationary and independent increments.
- B exhibits Gaussian increments

According to the paper, the drift ($\mu Stdt$) indicates the stock's general direction, while the shock ($\mu StdBt$) introduces random volatility.

According to the paper the other component of this model is the Drift. The drift is also known as the expected daily return of the stock, is calculated using the formula:

$$Drift = \mu - \frac{1}{2}\sigma^2 \text{ (2)}$$

According to the paper, μ is the mean of logarithmic returns, and σ^2 is the variance. Drift provides insights on the general direction of the stock so the past expected return of the stock is extrapolated into the future to forecast prices.

According to the paper, the other component of the model is the Shock. The shock is defined as the standard deviation of historical returns. It is calculated using the formula:

$$Shock = \sigma Z(Rand(0;1)) \text{ (3)}$$

According to the paper, σ is the standard deviation of the price, and Z is a random number, a simulated following a normal distribution. It determines the future volatility through a blend of historical deviation and a dash of randomness.

To weave it all together, we use the Monte Carlo Simulations. According to the paper, this algorithm utilizes randomness to approximate probable outcomes. Once the variables of the GBM model are determined, Monte Carlo Simulations will be performed according to the formula:

$$Price_i = Price_{i-1} \cdot e^{drift+shock} \quad (4)$$

$$Price_i = Price_{i-1} \cdot e^{\left(\mu - \frac{1}{2}\sigma^2\right) + \sigma Z(Rand(0;1))} \quad (4)$$

According to the ppaer, the likelihood of stock price following a given simulation calculated using the aforementioned formula is close to zero, therefore, Monte Carlo Simulations need to be repeated many times to determine the best fit line among all the simulations.

III. Data Collection and Treatment

For data collection for the project we obtain historical data for the S&P 500 and STOXX 50 indices from Yahoo Finance. The indices are representatives of the US and European markets, respectively. For the data retrieval process we use the yfinance library in Python. The symbols 'ĜSPC' and 'ŜTOXX50E' represent the S&P 500 and STOXX 50, respectively. The stock data collected was from January 1, 2015, to January 1, 2022. We also collected stock data from January 1, 2022 to January 1, 2023 to graph the true prices and compare how different they are from the predicted prices.

IV. Research Methodology

The financial data for S&P 500 and STOXX 50 is obtained from Yahoo Finance. Python, with libraries like NumPy, Pandas, Matplotlib, and yfinance, is used for data analysis. The GBM parameters are determined based on historical data, and Monte Carlo Simulation is performed to predict future stock prices. The Monte Carlo Simulation is run 10000 different times to get a generalized overall trend. The average of these future predictions is compared with the true prices to see how accurate the predictions are. The average is used to help mitigate the randomness and variability associated with individual simulations.

Python Code for Simulation

The following Python code illustrates the simulation process using Geometric Brownian Motion (GBM) for the S&P 500 and STOXX 50 indices:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import yfinance as yf
from datetime import datetime, timedelta

# Define the stock symbols for the US and European
    markets
us_symbol = '^GSPC'  # S&P 500 as a representative of
    the US market
european_symbol = '^STOXX50E'  # STOXX 50 as a
    representative of the European market

# Define the time period for historical data (adjust
    start and end dates as needed)
start_date = '2015-01-01'
end_date = '2022-01-01'

# Download historical data from Yahoo Finance
us_data = yf.download(us_symbol, start=start_date, end=
    end_date)
european_data = yf.download(european_symbol, start=
    start_date, end=end_date)

# Define the time period for historical data (adjust
    start and end dates as needed)
start_date = '2022-01-01'
end_date = '2023-01-01'

# Download historical data from Yahoo Finance
us_data_1 = yf.download(us_symbol, start=start_date,
    end=end_date)
european_data_1 = yf.download(european_symbol, start=
    start_date, end=end_date)
```

```python
# Function to simulate stock prices using GBM
def gbm_simulation(stock_data, num_simulations, mu, sigma, initial_price, num_days):
    returns = np.log(1 + stock_data['Adj Close'].pct_change())
    last_price = stock_data['Adj Close'].iloc[-1]

    simulation_df = pd.DataFrame()

    for i in range(num_simulations):
        price_series = [last_price]

        for j in range(num_days):
            price = price_series[-1] * np.exp((mu - 0.5 * sigma**2) + sigma * np.random.normal())
            price_series.append(price)

        simulation_df[f'Simulation_{i+1}'] = price_series

    return simulation_df

# Parameters for GBM and Monte Carlo Simulation
num_simulations = 10000
mu_us, sigma_us = np.mean(us_data['Adj Close'].pct_change()), np.std(us_data['Adj Close'].pct_change())
mu_european, sigma_european = np.mean(european_data['Adj Close'].pct_change()), np.std(european_data['Adj Close'].pct_change())

# Initial prices for simulation (adjust as needed)
initial_price_us = us_data['Adj Close'].iloc[-1]
initial_price_european = european_data['Adj Close'].iloc[-1]
```

```
# Number of days for the simulation
num_days = 360

# Perform simulations
us_simulations = gbm_simulation(us_data, num_simulations, mu_us, sigma_us, initial_price_us, num_days)
european_simulations = gbm_simulation(european_data, num_simulations, mu_european, sigma_european, initial_price_european, num_days)

# Calculate the average of the Monte Carlo simulations
us_simulations['Average'] = us_simulations.mean(axis=1)
european_simulations['Average'] = european_simulations.mean(axis=1)

# Plot True Prices for the US Stock Market
plt.figure(figsize=(12, 6))
plt.plot(us_data_1['Adj Close'].index, us_data_1['Adj Close'], label='True Prices', color='blue')
plt.title('True Prices for US Stock Market')
plt.xlabel('Days')
plt.ylabel('Stock Price')
plt.legend()
plt.show()

# Plot Monte Carlo Simulation Prices for the US Stock Market
plt.figure(figsize=(12, 6))
plt.plot(us_simulations, color='lightgray', linewidth=0.5)
plt.title('Monte Carlo Simulation Prices for US Stock Market')
plt.xlabel('Days')
plt.ylabel('Stock Price')
plt.show()
```

```
# Plot Average Monte Carlo Simulation Prices for the US
    Stock Market
plt.figure(figsize=(12, 6))
plt.plot(us_simulations['Average'], color='lightgray',
    linewidth=2, label='Average Simulation')
plt.title('Average Monte Carlo Simulation Prices for US
    Stock Market')
plt.xlabel('Days')
plt.ylabel('Stock Price')
plt.legend()
plt.show()

# Plot True Prices for the European Stock Market
plt.figure(figsize=(12, 6))
plt.plot(european_data_1['Adj Close'].index,
    european_data_1['Adj Close'], label='True Prices',
    color='blue')
plt.title('True Prices for European Stock Market')
plt.xlabel('Days')
plt.ylabel('Stock Price')
plt.legend()
plt.show()

# Plot Monte Carlo Simulation Prices for the European
    Stock Market
plt.figure(figsize=(12, 6))
plt.plot(european_simulations, color='lightgray',
    linewidth=0.5)
plt.title('Monte Carlo Simulation Prices for European
    Stock Market')
plt.xlabel('Days')
plt.ylabel('Stock Price')
plt.show()

# Plot Average Monte Carlo Simulation Prices for the
    European Stock Market
plt.figure(figsize=(12, 6))
```

```
plt.plot(european_simulations['Average'], color='
    lightgray', linewidth=2, label='Average Simulation')
plt.title('Average Monte Carlo Simulation Prices for
    European Stock Market')
plt.xlabel('Days')
plt.ylabel('Stock Price')
plt.legend()
plt.show()
```

V. Results and Discussion

The Monte Carlo Simulation with 10000 simulations was run for both the US and European markets. These simulations suggest market trends, and true prices are compared with simulated prices to evaluate the accuracy of the model.

True Prices Plot for US Stock Market

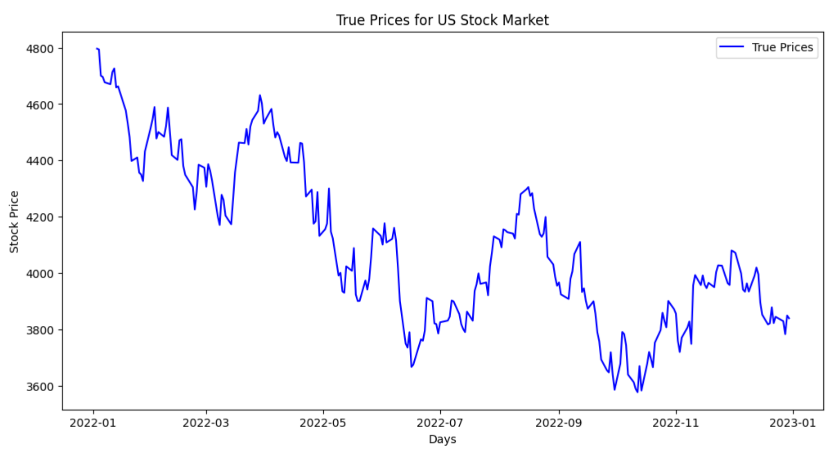
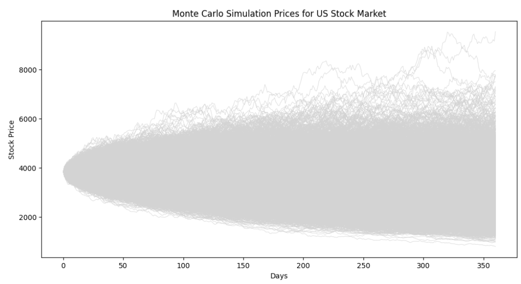


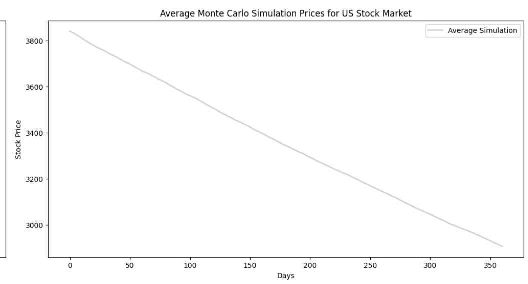
Figure 1: True Prices for US Stock Market

The plot in Figure 5 illustrates the true prices of the US stock market over January 1, 2022 to January 1, 2023.

Monte Carlo Simulation Prices Plot for US Stock Market



(a) Monte Carlo Simulations for US Stock Market



(b) Average of Monte Carlo Simulations for US Stock Market

Explanation

The plot in figure a) shows the 10000 different Monte Carlo simulations run using stock data from the S&P 500 from January 1, 2015 to January 1, 2020. The plot in figure b) is the average of the 10000 simulations shown in figure a). As we can see in Figure 1 and Figure b), the predicted stock price generated by the Monte Carlo method is not the same as the actual stock price. The price difference in the stock price between the start of the year to the end of the year as shown by Figure 1 is approximately 4800-3900 = 900 points. The price difference in the stock price between the start of the year to the end of the year as shown by Figure b is approximately 3800-2900 = 900 points. So the Monte Carlo approximation is able to measure the difference in stock price at the start of the year to the end of the year quite accurately.

True Prices Plot for European Stock Market

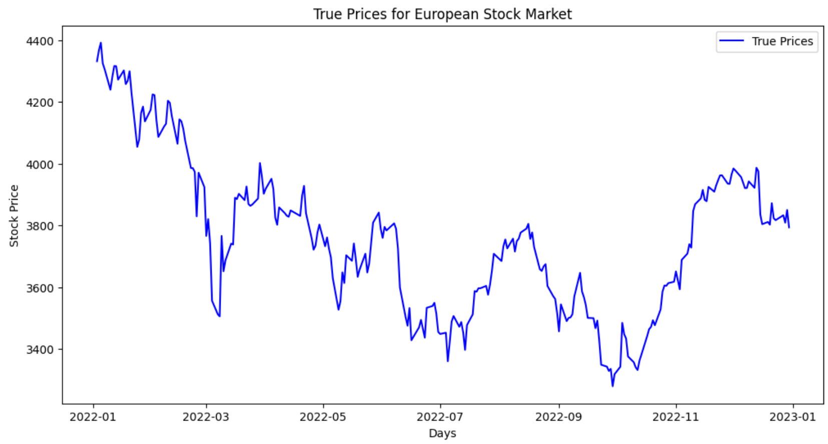
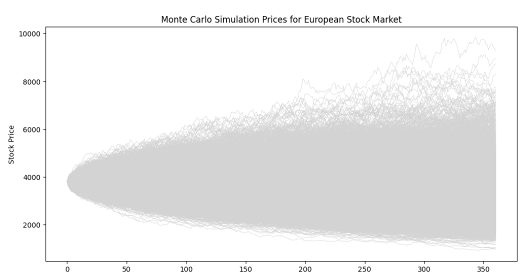


Figure 3: True Prices for European Stock Market

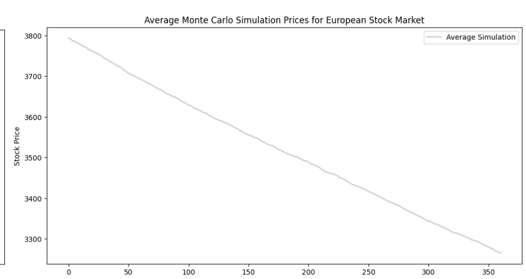
Explanation

The plot in Figure 7 illustrates the true prices of the European stock market over January 1, 2022 to January 1, 2023.

Monte Carlo Simulation Prices Plot for European Stock Market



(a) Monte Carlo Simulations for European Stock Market



(b) Average of Monte Carlo Simulations for European Stock Market

Explanation

The plot in figure a) shows the 10000 different Monte Carlo simulations run using stock data from the STOXX 50 from January 1, 2015 to January 1, 2022. The plot in figure b) is the average of the 10000 simulations shown in figure a). As we can see in Figure 1 and Figure b), the predicted stock price generated by the Monte Carlo method is not the same as the actual stock price. The price difference in the stock price between the start of the year to the end of the year as shown by Figure 1 is approximately 4400-3800 = 600 points. The price difference in the stock price between the start of the year to the end of the year as shown by Figure b is approximately 3800-3250 = 550 points. So the Monte Carlo approximation is able to measure the difference in stock price at the start of the year to the end of the year quite accurately.

a. Assumption from above data

From the above data it seems that the Monte Carlo Simulation is good at predicting the point difference between the stock price at the beginning of the year and the stock price at the end of the year. To test this assumption the code was modified so that the Monte Carlo Simulation used stock data from January 1, 2015 to January 1, 2023. The true prices of stock from January 1, 2023 to January 1, 2024 were recorded for checking accuracy of the Monte Carlo Simulation. This was done for both the S&P 500 and the STOXX 50 indices.

b. Observations

True Prices Plot for US Stock Market

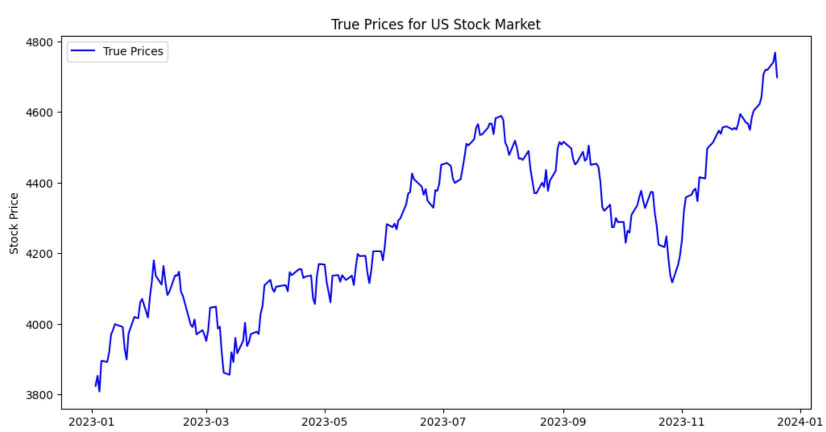
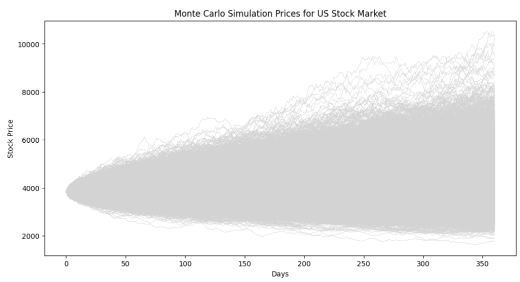


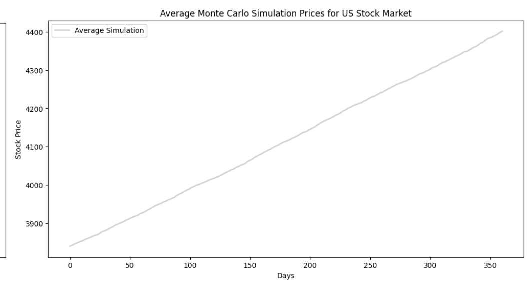
Figure 5: True Prices for US Stock Market

The plot in Figure 5 illustrates the true prices of the US stock market over January 1, 2023 to January 1, 2024.

Monte Carlo Simulation Prices Plot for US Stock Market



(a) Monte Carlo Simulations for US Stock Market



(b) Average of Monte Carlo Simulations for US Stock Market

Explanation

The plot in figure a) shows the 10000 different Monte Carlo simulations run using stock data from the S&P 500 from January 1, 2015 to January 1, 2023. The plot in figure b) is the average of the 10000 simulations shown in figure a). The price difference in the stock price between the start of the year to the end of the year as shown by Figure 1 is approximately 4800-3800 = 1000 points. The price difference in the stock price between the start of the year to the end of the year as shown by Figure b is approximately 4400-3850 = 550 points. So the Monte Carlo approximation is unable to measure the difference in stock price at the start of the year to the end of the year quite accurately.

True Prices Plot for European Stock Market

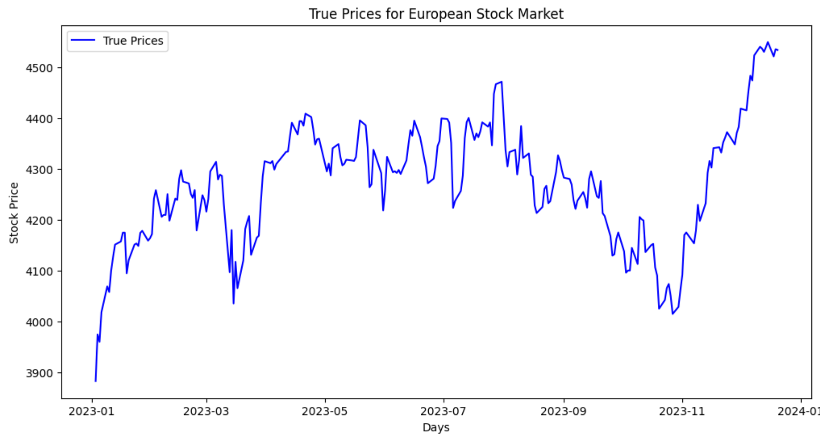
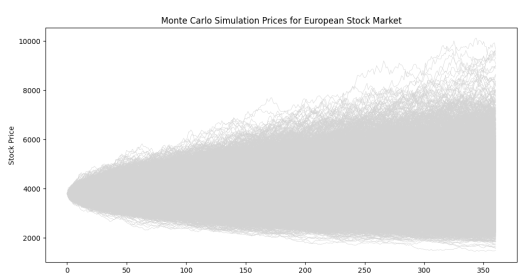


Figure 7: True Prices for European Stock Market

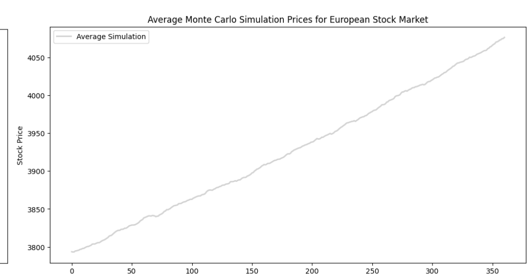
Explanation

The plot in Figure 7 illustrates the true prices of the European stock market over January 1, 2023 to January 1, 2024.

Monte Carlo Simulation Prices Plot for European Stock Market



(a) Monte Carlo Simulations for European Stock Market



(b) Average of Monte Carlo Simulations for European Stock Market

Explanation

The plot in figure a) shows the 10000 different Monte Carlo simulations run using stock data from the STOXX 50 from January 1, 2015 to January 1, 2023. The plot in figure b) is the average of the 10000 simulations shown in figure a). The price difference in the stock price between the start of the year to the end of the year as shown by Figure 1 is approximately 4550-3900 = 650 points. The price difference in the stock price between the start of the year to the end of the year as shown by Figure b is approximately 4075-3800 = 275 points. So the Monte Carlo approximation is unable to measure the difference in stock price at the start of the year to the end of the year quite accurately.

c. Observations from data

From the above data, we can conclude that the Monte Carlo Simulation is unable to predict actual stock price in the future. It is also unable to predict the actual point difference between the stock prices at the beginning of the year and the end of the year. However, the Monte Carlo Simulation is able to predict whether the stock value will increase or decrease in the next year and it is also able to tell which stock will increase more in value compared to the other. From January 1, 2023 to January 1, 2024, the US stock prices increased by 1000 points whereas the European stock prices increased by 650 points. The Monte Carlo Simulation for US stock prices increased by 550 points where as the Monte Carlo Simulation for European stock prices increased by 275 points. Hence, the Monte Carlo Simulation was able to predict that the US stock prices would increase more as compared to the European stock prices.

VI. Limitations of Method

As seen above the models used are not accurate in predicting prices. Some of the limitations of financial modeling using Monte Carlo Simulation and Geometric Brownian Motion (GBM) modeling could be:

- Market Dynamics and Assumption of Stationarity: Financial markets are dynamic and subject to changing conditions. If the model assumes constant statistical properties over time, it may not fully capture the non-stationary nature of financial data.

- Data Limitations and Historical Bias: The accuracy of predictions is contingent on the quality and representativeness of historical data.

Limitations in data quality, availability, or historical bias can introduce uncertainties.

- Assumption of Normality: The GBM model often assumes that asset returns follow a normal distribution. In reality, financial markets may exhibit non-normal behavior, especially during extreme events. Highlighting the limitations associated with the normality assumption adds transparency to the model's assumptions.
- Overlooking External Factors: Monte Carlo Simulation and GBM modeling may not fully account for external factors, such as geopolitical events, regulatory changes, or sudden market shocks.
- Computational Intensity: Performing Monte Carlo Simulations can be computationally intensive, especially when running a large number of simulations.

VII. Conclusion

In conclusion, this project helps understand of the strengths and limitations of Monte Carlo Simulation and GBM modeling in the context of financial forecasting. Through this project we explored the application of Monte Carlo Simulation and Geometric Brownian Motion (GBM) modeling for predicting stock prices in the US and European markets. The analysis involved utilizing historical stock data, defining GBM parameters, and conducting Monte Carlo Simulations to forecast future prices. These Monte Carlo Simulations provided valuable insights into potential market trends, and the comparison with true prices helped assess the model's accuracy. The Monte Carlo Simulation was able to capture directional trends, indicating whether stock prices were likely to increase or decrease. However, it was unable to predict actual stock prices or point differences.

VIII. References

J. N. P. Xiang, S. R. Velu and S. Zygiaris, "Monte Carlo Simulation Prediction of Stock Prices," 2021 14th International Conference on Developments in eSystems Engineering (DeSE), Sharjah, United Arab Emirates, 2021, pp.

212-216, doi: 10.1109/DeSE54285.2021.9719349.
https://ieeexplore.ieee.org/document/9719349