Applied Computer Science, vol. 19, no. 3, pp. 64–82
doi: 10.35784/acs-2023-25

Submitted: 2023-07-04 | Revised: 2023-09-13 | Accepted: 2023-09-14

Keywords: Stock Trading Algorithm, Machine Learning, SVM, Performance Analysis

Tilla IZSÁK [0009-0002-4275-5279]*, László MARÁK [0000-0002-2280-8014] **
Mihály ORMOS [0000-0002-3224-7636]*

# EVALUATION OF STOCK PRICE PREDICTION BASED ON THE SUPPORT VECTOR

**Abstract**

*In recent years with the advent of computational power, Machine Learning has become a popular approach in financial forecasting, particularly for stock price analysis. In this paper, the authors develop a non-recurrent active trading algorithm based on stock price prediction, using Support Vector Machines on high frequency data, and compare its risk adjusted performance to the returns of a statistical portfolio predicted by the Capital Asset Pricing Model. The authors selected the three highest volume securities from a pool of 100 initially selected stock dataset to investigate the algorithmic trading strategy. The abnormal return estimates are significant and positive, and the systematic risk is lower than unity in all cases, suggesting lower risk compared to the market. Moreover, the estimated beta values for all stocks were close to zero, indicating a market independent process. The correlation analysis revealed weak correlations among the processes, supporting the potential for risk reduction and volatility mitigation through portfolio diversification. The authors tested an equally weighted portfolio of the selected three assets and demonstrated a remarkable return of 1348% during the evaluation period from July 1st, 2020, to January 1st, 2023. The results suggest that the weak form of market efficiency can be questioned, as the algorithmic trading strategy, employing a Support Vector Machine binary classification model, has consistently generated statistically significant and substantial abnormal returns using historical market data.*

## 1. INTRODUCTION

The prediction of stock prices is a challenging task that has been of great interest to investors, traders, and researchers alike. Fama (1971, 1991) argues that, as stock prices reflect all available information, they are in equilibrium; whenever added information arrives, the rational arbitrageurs are immediately built into the prices; thus, the prices

* J. Selye University, Faculty of Economics and Informatics, Department of Economics, Slovakia, izsak.tilla@student.ujs.sk, ormosm@ujs.sk
** J. Selye University, Faculty of Economics and Informatics, Department of Informatics, Slovakia, marakl@ujs.sk

cannot be predicted using past information. However, this research shows that with the advent of Big Data and Machine Learning techniques, it has become possible to develop models that can accurately predict trends in stock prices.

In recent years, various Machine Learning algorithms have been applied to stock price prediction with promising results (Vijh, 2020). However, the authors in these studies frequently use datasets consisting of Open, High, Low, Close, Volume values (OHLCV) (e.g., Ariyo, 2014). These datasets are generally and readily available, which might be the reason behind their popularity. Unfortunately, as these values in generally available datasets are already aggregated, they cannot provide information on individual trades or the specific moments at which those trades occurred. This limits the accuracy of the predictions made using these datasets. Indeed, it is intuitively justified that direct, high resolution data can provide better insight into the inner workings of price development.

The stock market is a complex system that is affected by a wide range of factors, including economic indicators, company performance, and global events as well as countless momentary effects such as: take-off, first sell off, bear trap, media attention, enthusiasm, greed, delusion, new paradigm, denial, mean reversion, fear, capitulation, or despair (Jacobs, 1989). Traditional asset pricing methods like the Capital Asset Pricing Model (CAPM) are widely used for estimating the expected return of an investment based on the market's expected return and the asset's risk (Fama - French, 2004), and are widely used for determining expected returns on an investment (Acharya, 2005). Due to its design, CAPM applies a single variable, the market-beta, to represent the risk of an asset. In Machine Learning, on the other hand, we can handle high-dimensional and non-linear data, making it a well-suited approach for high-throughput data prediction, such as the stock market.

The primary objective of this paper is to challenge the weak form of market efficiency by employing SVM machine learning technique for stock price (movement) prediction. The principal research gap for which this study fills in is the comparison of a non-recurrent machine learning algorithm-based trading strategy with a Balanced Static Portfolio. The hypothesis of the study is that a Machine Lea algorithm can result in lower level of risk compared to a Balanced Static Portfolio, as predicted by the Capital Asset Pricing Model (CAPM). To achieve this goal, the authors conducted a comprehensive examination of the application of Support Vector Machines (SVM) in stock price prediction, comparing its performance against the traditional CAPM model. Additionally, the paper discusses key factors contributing to the superior performance of the algorithm, emphasizing the significance of the feature selection, data pre-processing, and model selection in achieving optimal trading performance.

The research question is: "Based on the CAPM predictions, can a non-recurrent trading algorithm lead to higher returns with respect to the risk than a Balanced Static Portfolio?”. This study compares the prediction algorithm with the CAPM and provides insight into the factors that are most important for prediction accuracy. The results show that it is possible to significantly beat the market using the SVM algorithm.

This paper is organized as follows: the opening section offers an overview of previously published research concerning stock market price prediction. The subsequent section introduces the theoretical foundations of the Machine Learning technique employed, specifically Support Vector Machines (SVM), alongside the presentation of our trading strategy and the CAMP model. The second section describes the methodology of the authors' algorithm, encompassing the data collection process, the selected instruments, the parameter optimization, and the model training. The third section entails a comprehensive exposition and analysis of the simulation results (Fig.1). In the fourth section the authors evaluate the performance of both models on a dataset of historical stock prices and demonstrate that Machine Learning models outperform the CAPM in terms of prediction accuracy. Additionally, they discuss the potential reasons for this superiority and highlight the importance of proper feature selection, data pre-processing, and model selection in achieving high performance with the Machine Learning models. Finally, the fifth section presents the results which suggest that Machine Learning is a promising approach for stock price prediction and offer insights for future research in this area.

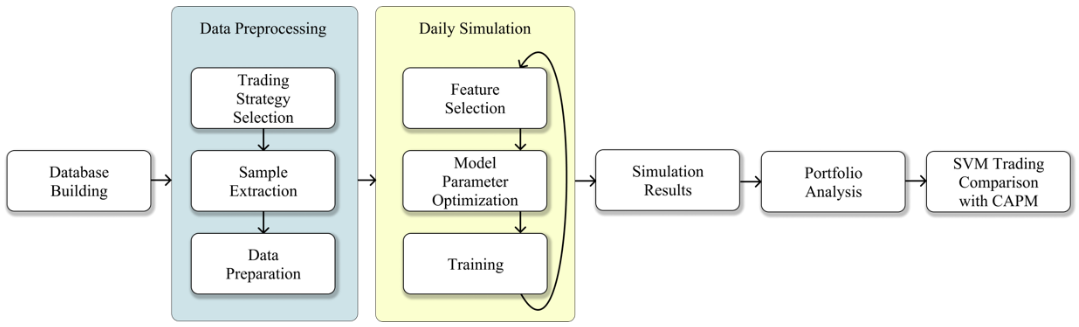

**Fig. 1. The Conceptual Model**

## 2. THEORETICAL BACKGROUND

### 2.1. Literature Review

The existing literature on the prediction of high-frequency stock market metadata has been the subject of extensive research in recent years. Various approaches have been proposed to forecast stock market movements and trends using machine learning techniques. Several studies have delved into this area, employing distinct datasets, methodologies, and specialized focuses. Notably, the literature review reveals that these research efforts predominantly utilize one or more parameters from the open, high, low, close, and volume (OHLCV) data, along with a selection of additional single-variable parameters. For instance, Lu et al. (2021), Zhang and Khushi (2021), Ji et al. (2021), Yu and Yan (2020), and Xu and Zhang (2023) have each contributed to this field. Lu et al. (2021) specialized in analyzing Chinese stocks, whereas Zhang and Khushi (2021) focused on the FOREX market, utilizing OHLCV data aggregated at a 5-minute interval. Furthermore, it is notable that some researchers, such as Lai et al. (2023), Briola et al. (2021), and Kohda and Yoshida (2022), have also explored the utilization of the Limit Order Books (LOBs) in their research. Lai et al. (2023), for example, employed anonymized datasets for their research, which posed challenges for testing their trained models on live data. Additionally, they conducted live tests on trading data from the Chinese Stock Exchange but limited their analysis to a brief 7 and 10-day period. In contrast, our research takes a distinctive approach. We have conducted experiments over an extended period of 2.5 years, seeking to demonstrate the consistent performance of our prediction algorithm over a more substantial period. This extended evaluation allows us to offer insights into the algorithm's robustness and adaptability in various market conditions. While previous studies have laid a foundation for predicting stock market metadata, our research contributes by providing evidence of the algorithm's long-term effectiveness, potentially offering more reliable insights for investors and market analysts.

## 2.2. Support Vector Machine

The Support Vector Machine (SVM) algorithm is a supervised learning model and the associated learning algorithm used in machine learning. It is used for classification and regression analysis by analyzing high dimensional data. The SVM has been developed at AT&T Bell Laboratories by Vladimir Vapnik and his colleagues (Vapnik, 1995). The SVM has been used in numerous studies to predict stock prices (Kim, 2003), (Henrique, 2018), (Sapankevych, 2009). It is a type of supervised learning algorithm that can be used for binary classification problems, among other tasks. In SVM, data points are represented as vectors in a multi-dimensional space, and the objective is to determine if the data points can be separated by a hyperplane. The goal is to find the hyperplane that maximizes the margin, which is the distance between the hyperplane and the closest data points from each class. The SVM simultaneously minimizes the empirical classification error and maximizes the geometric margin. During the training SVM is searching for a separating hyperplane that separates the points of the two classes. It is assumed that the larger the margin or distance between the hyperplanes the better the generalization error of the classifier. If we represent the training data $D$ as follows:

$$D \coloneqq \{(\underline{x}_1, y_1), (\underline{x}_2, y_2), \dots, (\underline{x}_n, y_n)\} \tag{1}$$

where $y_i \in \{1, -1\}, i \in 1 \dots n$ is a constant representing the class to which the $\underline{x}_i$ point belongs and $n$ denotes the number of samples. Each $\underline{x}_i$ is a $p$-dimensional real vector. Scaling is important to protect the features (attributes) with higher variances. This training data is obtained using the separating hyperplane given by $\underline{w} \cdot \underline{x} + b = 0$, where $b$ is a scalar and $\underline{w}$ is a $p$-dimensional vector. The $\underline{w}$ vector is perpendicular to the separating hyperplane. The inclusion of the parameter $b$ allows for increasing the margin. Without $b$, the hyperplane would have to pass through the origin, limiting the solution. The parallel hyperplanes can be defined as follows:

$$\underline{w} \cdot \underline{x} + b = 1 \qquad \text{and}$$
$$\underline{w} \cdot \underline{x} + b = -1$$

The result of the training $T$ is a subset of the training data ($T \subset D$). The samples $\underline{x}_i \in T$ are along the margins and are called Support Vectors. As the Support Vectors represent the samples along the separating hyperplane, the hyperplane can be expressed as the superposition of the Support Vectors

$$\underline{w} = \sum_{\underline{x_i} \in T} \underline{x_i} \tag{2}$$

The hyperplane with the largest margin, defined by $M = \frac{2}{|w|}$, will satisfy the following equation:

$$y_i(\underline{w} \cdot \underline{x_i} + b) = 1, \forall i \tag{3}$$

where $\underline{x_i} \in T$ is a Support Vector, and for an optimal hyperplane

$$y_j(\underline{w} \cdot \underline{x_j} + b) \geq 1, \forall j \in 1..n \tag{4}$$

where $n$ is the number of training data points. To find the optimal hyperplane with maximum margin, during the training process, the model needs to minimize $\|w\|_2$ subject to inequality constraints (4). This optimization problem can be solved using the Lagrange primal-dual optimization method.

In many cases, however, there is no optimal $n-1$ dimensional hyperplane separating the sample points of the training data. To separate the samples, we can use a "kernel trick". The "trick" consists of replacing the inner product operator with an inner product function $K$ of a higher dimensional space. Indeed, the kernel space can even be infinite dimensional. Equation (4) becomes:

$$y_j\left(\sum_{\underline{x_i} \in T} K\left(\underline{x_i}, \underline{x_j}\right) + b\right) \geq 1, \forall j \in 1..n \tag{5}$$

We can imagine the method as embedding the sample vectors into the higher dimensional space. After the embedding, we are looking for the optimal hyperplane in the broader space.
The most used kernel function is the Radial Basis Function (RBF) kernel function:

$$K(\underline{x_i}, \underline{x_j}) = e^{-\gamma\left\|\underline{x_i}-\underline{x_j}\right\|^2} \tag{6}$$

Where $\gamma > 0$ is the parameter of the kernel. The characteristic of the RBF kernel is that it nonlinearly maps the samples to a higher-dimensional space. Its usage is widely spread in various problem scenarios (Durgesh, 2010). In our case, the SVM model is used as a binary classifier which classifies the price movement as either favorable (up) or unfavorable (not-up). The training "tries to identify" patterns in the data that would indicate a sudden rise in the price of the particular instrument.

## 2.3. Trading Strategy

The authors develop a predictive model that aims to perform day trading by utilizing a live market data stream. The trading strategy is as follows. The algorithm is analyzing the stream of current instrument prices. If the algorithm predicts a rise in the instrument price by $k$ percent within the next $t$ seconds, then it indicates a buy. Here $k$ and $t$ are predefined constants determined before the training and $t$ acts as a rolling window. Since we want to build a binary classification model, we need to have two types of classes. The positive class, which would indicate a significant rise in the price of the instrument, and the negative class, where the model detects no indication of sudden rise. As a safety measure, we have implemented a stop-loss function, which is activated when the price begins to fall against the predictions within the time window. The stop-loss limits potential losses by selling the stock if the price falls below a predefined limit (usually 1 percent below the buying price). Our trading strategy consists of exclusively of day trading, which means that we must buy and sell the instrument on the same day. Our goal is to find moments where the exchange rate would likely increase by at least $k$ percent within $t$ seconds. To collect the samples for the training, we use historical data that we have collected using a publicly accessible trading platform (Interactive Brokers).

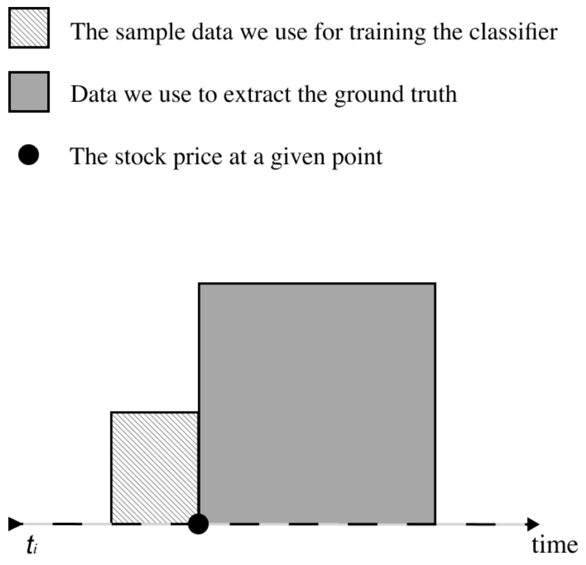
▨ The sample data we use for training the classifier
■ Data we use to extract the ground truth
● The stock price at a given point
$t_i$
time

**Fig. 2. Methodology for Creating the Trading Samples**

## 2.4. Performance Measure

The authors use the standard Capital Asset Pricing Model (CAPM) to measure the risk adjusted return and abnormal performance of the strategy generated by the SVM. CAPM is a theoretical model for calculating the expected return of an investment, given the risk-free rate, the expected return of the market, and the assets' Beta (a measure of the price process' sensitivity to the market portfolio, Sharpe, 1964). The model was first proposed by Sharpe in 1964 and later refined by other economists, including Treynor (1965), Mossin (1966) and Lintner (1969). The expected market rate of return is evaluated through the computation of the arithmetic mean of historical returns on a market portfolio. The Alpha is also calculated, which is essential for determining the abnormal return of an instrument or portfolio compared to the theoretical expected return (Jensen, 1968). This study employs the S&P 500 index for this purpose. The CAPM index model with alpha $n_d$ error is the following:

$$R_i = R_f + \beta_i \times (R_m - R_f) + \alpha_i \pm \varepsilon_i \tag{7}$$

| | | |
|---|---|---|
| where: | $R_i$ | is the actual return of asset $i$, |
| | $R_f$ | is the risk-free rate of return, |
| | $\beta_i$ | is the Beta coefficient of asset $i$, |
| | $R_m$ | is the return of the market portfolio, |
| | $\alpha_i$ | is the Alpha, which represents the abnormal return of asset $i$, and |
| | $\varepsilon_i$ | is the error term, which captures any unexplained or random variation. |

In our calculations we have used the following values:

| | |
|---|---|
| $R_f =$ | 0.017824 |
| $\beta_i =$ | 1.9593 |
| $R_m =$ | 0.380122 |
| $\alpha_i =$ | 12.333341 |
| $\varepsilon_i =$ | 0.533490 |

$R_i$= 0.017824 + 1.9593*(0.380122 - 0.017824) +12.333341 + 0.53349 = 1359.452%
$R_i$= 0.017824 + 1.9593*(0.380122 - 0.017824) +12.333341 - 0.53349 = 1252.754%

The authors conducted an analysis using the Capital Asset Pricing Model (CAPM) index model, incorporating the Alpha and the Error over a specific time interval of two and a half years. The analysis focused on a portfolio comprising of three stocks. The findings of the study indicate that the expected return within this interval falls within the range of 1252% to 1359%. Comparing these results with those of the algorithmic trading strategy under study, specifically in relation to the return generated by the equally weighted portfolio (1348%), it can be concluded that the return projected by the CAPM index model for the examined period aligns closely with the actual return achieved through the algorithmic trading approach under study.

## 3. METHODOLOGY

### 3.1. Selecting the Instruments for the Study

The dataset utilized in this study has been procured from a source containing high resolution market data. Electronic access to this market data was facilitated through Interactive Brokers, a brokerage firm. The authors used the IBKR Application Programming Interface (API) to obtain real-time, non-aggregated market data. They collected this data and constructed a high temporal resolution database containing realistic market data, with data collection commencing on July 1, 2020. It is important to emphasize that this database is non-reproducible due to its inclusion of real-time market metadata that was acquired and stored in real-time fashion.

The dataset consists of approximately 100 instruments, which have initially been selected based on financial analysts' predictions as having the potential for improvement. After the initial evaluation it was concluded that the instruments with high trading volume responded best to the training algorithm, and therefore the authors selected the three instruments with the highest trading volume for this study. We suppose that the reason behind the model's superior performance was the high temporal resolution of the data. The companies selected for the study were AMD (Advanced Micro Devices, Inc.), NVDA (NVIDIA Corporation) and (Roku, Inc.). AMD is a semiconductor company that specializes in designing and producing microprocessors, graphics processing units (GPUs) and other computer components. NVIDIA is also a technology company that designs and produces GPUs and AI computing technology. NVIDIA's GPUs are widely used in various applications, including gaming, professional visualization, and scientific research. Roku is a streaming platform that provides access to various forms of content, including movies, TV shows, and live sports.

### 3.2. Summary of our Trading Algorithm

A summary of the main steps of the trading algorithm:

- **Data Collection and Preparation:** The first step is to gather and prepare (Uniform time series sampling) the data for the model. The data should be labeled based on our trading strategy, and then split into a training dataset and a test dataset.
- **Feature Selection:** The next step is to select the features that will be used in the model. This is a key step as it can affect the performance of the model (Ding, 2005).
- **Model Training:** The SVM model is trained using the preprocessed training dataset.
- **Parameter Tuning:** Optimizing the model's performance by tuning the model's parameters. The authors evaluate the pre-trained model using a part of the training dataset. The accuracy of the model is measured by comparing the predicted labels to the ground truth and we select the optimal training parameters based on the model performance.
- **Simulation:** Evaluating the performance of the final model on the test dataset. The test dataset consists of the most recent prices of the instruments and has neither been used for the training nor for the parameter tuning.
- After the simulation the authors compare the results with the generally available CAPM returns.

### 3.3. Data Collection

The authors retrieved the tick data for specific instruments, including the time, price, and other information about each trade. Tick data, also known as "time and sales data," is a record of every transaction that occurs on a financial market. The next step is to continuously examine the inspected instrument's price changes and find samples for each class using the trading strategy explained in section Trading Strategy. After identifying such moments, we collect data for the corresponding samples (positive or negative). The tick types which we have chosen to use in our SVM model are:

- Bid Price – Highest priced bid for the contract,
- Ask Price – Lowest price offer on the contract,
- Last Price – Last price at which the contract was traded,
- Market Price – The market price is the current theoretical calculated value of an instrument,
- Close Price – The last available closing price for the previous day,
- Open Tick – Current session's opening price.
- Low 13 Weeks – Lowest price for the last 13 weeks (about 3 months),
- High 13 Weeks – Highest price for the last 13 weeks (about 3 months),
- Low 26 Weeks – Lowest price for the last 26 weeks (about 6 months),
- High 26 Weeks – Highest price for the last 26 weeks (about 6 months),
- Low 52 Weeks – Lowest price for the last 52 weeks (about 12 months), and
- High 52 Week – Highest price for the last 52 weeks (about 12 months).

The first three features (Bid Price, Ask Price, Last Price) are represented as a time series format in the sample (each consists of data collected over a 5-minute interval), the rest are represented as one parameter each. We have collected data for the training and testing sets using a slightly different approach. The training set consisted of positive and negative samples in approximately equal proportions, while the testing dataset had a ratio of negative samples exceeding 90 percent and positive samples less than 10 percent. The 10%-90% ratio allowed us to strike a balance between comprehensiveness and realism in our testing approach, facilitating a more meaningful evaluation of the algorithm's performance under common market conditions. Moreover, we have created training and testing samples from the entire examined time interval, which spanned over two years. Since we were conducting day trading, we aimed to test our model over the maximum timeline.

## 3.4. Sampling Technique

The main characteristics of the time series data is that each tick is created in the moment of the trade, which means the time data is not uniform. The ticks are collected and recorded over time and may differ in sampling intervals through time. To analyze this data, it may be necessary to select a subset of the data. Uniform time series sampling is a statistical method of selecting a sample from a time series dataset in a way that ensures that the sample is representative of the entire dataset. The technique is based on selecting a fixed number of observations at regular intervals from the time series data. In our case, we used a time series data interval of 5 minutes. Since the density of stock market trading varies, the amount of tick data received in each time interval also varies, we have therefore represented the data with a data series of 300 data points per 5 minutes, meaning with a sampling density of 1 sample per second.

## 3.5. Data Transformation and Feature Selection

The objective of data normalization is to ensure that all variables are on the same scale and have comparable values. The authors performed a two-step data normalization during the data preparation phase of their Machine Learning model. The first normalization step unified the prices of the instruments across the samples. As we were interested in the changes of the instrument price, we scaled the initial price of the instrument at each time interval sample to one. This way we have removed the absolute price of the instrument from the sample and only the relative price increase or decrease remained in the sample. In the second step of the normalization, we have employed a vertical standard score normalization on the relative prices from the first step in addition to the remaining 11 singular features.

For feature selection and dimensionality reduction we used the Minimum Redundancy Maximum Relevance (MRMR) feature selection algorithm. The MRMR algorithm identifies an optimal subset of features that are highly dissimilar to each other and can efficiently represent the response variable. The algorithm aims to minimize redundancy in the feature set while maximizing its relevance to the response variable. Mutual information is used to quantify the redundancy and relevance of the features, including the pairwise mutual information of the features and the mutual information between a feature and the response (Ding, 2005).

## 3.6. Validation techniques and Hyper-Parameter Optimization

Validation techniques were used to evaluate the performance of the Support Vector Machine (SVM) binary classification model to ensure that it generalizes well to unseen data. A nested holdout method was used to validate the data. This method uses both hold-out validation and Bayesian parameter optimization. It divides the training data into a training set and a test set. The authors train the SVM model on the training set and then evaluate it on the test set. The outer loop performs the hold out-validation, while the inner loop performs the Bayesian optimization to optimize the models' parameters. The accuracy of the model is measured by comparing the predicted labels to the true labels. The Bayesian Optimizer is a method for finding the optimal values of hyper-parameters for a Machine Learning model. The Bayesian optimizer works by defining a probability distribution over the possible values of the hyper-parameters (Snoek, 2012). This distribution is called the prior distribution. The model is updated as added information becomes available (i.e., the performance of the model on the training data). This updated distribution is called the posterior distribution. The Bayesian optimizer then samples from the posterior distribution to select the next set of hyper-parameters to use. This process is repeated until a stopping criterion (objective function) is met, or a maximum number of iterations is reached. An objective function is a mathematical function that is used to evaluate the performance of the optimizer. The function takes a set of input parameters and returns a single scalar value that represents the "goodness" of the solution. The goal of optimization is to find the set of input parameters that maximize the objective function. In Machine Learning the objective function could be the accuracy of a model on a validation set, and the input parameters could be the SVM parameters of the training. The goal of the

optimization would be to find the set of hyper-parameters that result in the highest accuracy.
The following parameters were optimized:

- Kernel (Linear or RBF) – SVM model parameter
- Kernel scale – SVM model parameter
- Regularization parameter (Box Constraint) – SVM model parameter
- Cost Parameter – SVM model parameter
- The size of the training set – Data sample parameter
- The number of selected features – Data sample parameter

### 3.7. Training

The authors collected the data for this study for the period of two and a half years from 1st July 2020 until 1st January 2023. After the sample collection and data preprocessing, two datasets were created, one for training and one for testing (Table 1). In Machine Learning, it is crucial to have a balanced dataset for training, which means that the number of positive and negative samples should be roughly equal. This is to avoid bias towards either class during the training process, which can lead to poor performance on the test set. In this study, the training dataset was carefully designed to have an equal distribution of positive and negative samples at a ratio of 50-50 percent. However, it is important to note that in real-world scenarios, the proportion of positive and negative samples is often imbalanced. To test the performance of the model on a more realistic dataset, the testing dataset was designed to reflect a more imbalanced distribution of positive and negative samples. The testing dataset had a ratio of approximately 90% of negative samples and 10% of positive samples. This allowed to evaluate the model's ability to handle imbalanced datasets.

**Tab. 1. Training and Testing datasets**

| Instruments | Training Samples | Testing Samples |
|---|---:|---:|
| AMD | 22 197 | 197 315 |
| NVDA | 22 610 | 194 175 |
| ROKU | 26 330 | 81 484 |
| The distribution of the sample classes | 50% – positive<br>50% – negative | 10% – positive<br>90% – negative |

## 4. EVALUATION OF THE TRADING STRATEGY

The final model was evaluated using the original test set, which was not used during training. We have chosen a single day (exclusively business days) as the testing period for the model. One day (business days only) was chosen as the test period for the model. The training tested is chronological, meaning that the authors train the model on past data, and it predicts on "new" data. The simulation was repeated daily for the two and a half year-period. The model’s predictions were used as indicators to either buy (positive) or abstain from buying (negative) the target instrument. In order to approach real scenarios, the simulation was initiated with an initial capital of $10,000 (ten thousand dollars), and a transaction cost of $1 was used for each purchase, both for buying and selling. A strategy was implemented in which all available capital was used for each purchase. When the model recommended a buy, all available capital for purchasing the instrument was spent. Also, as a general rule, the authors did not commence another transaction before the previous position had been closed. The simulation produced favorable results, with each stock being evaluated separately with an initial budget of $10 000, and all three stocks were exhibiting a profitable performance during the simulation period. By the end of the two-and-a-half-year period, the budget allocated to AMD had increased by nearly 2,000% (Figure 3), NVDA had grown by 1,400% (Figure 4), and ROKU had increased by 600% (Figure 5).

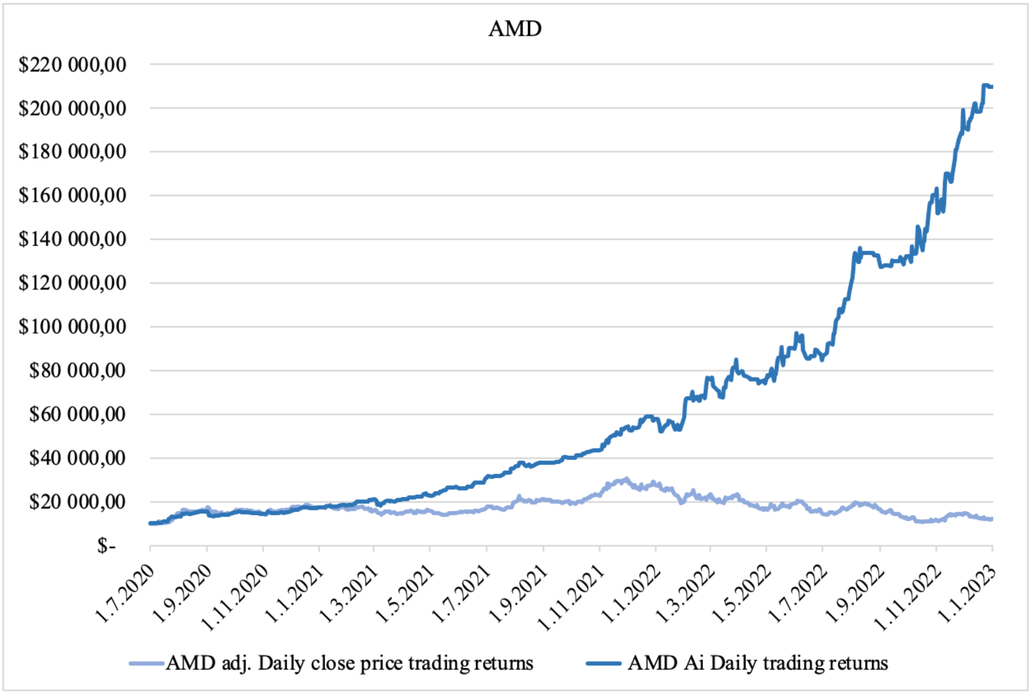

**Fig. 3. Ai trading results for AMD compared with the Instrument Price Chart (Adj. Close Price) between 1st July 2020 and 1st January 2023**

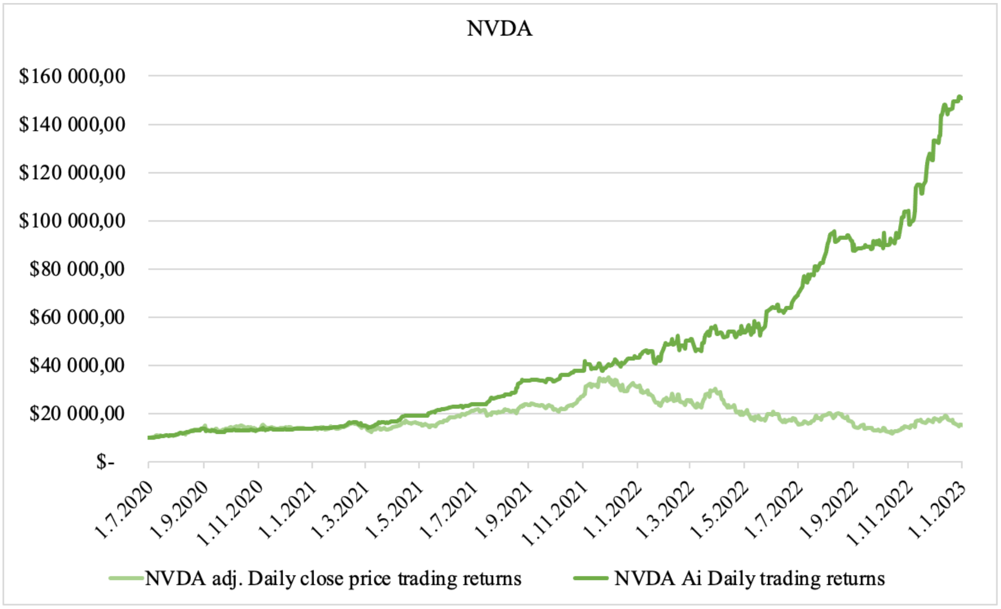

**Fig. 4. Ai trading results for NVDA compared with the Instrument Price Chart (Adj. Close Price) between 1st July 2020 and 1st January 2023**

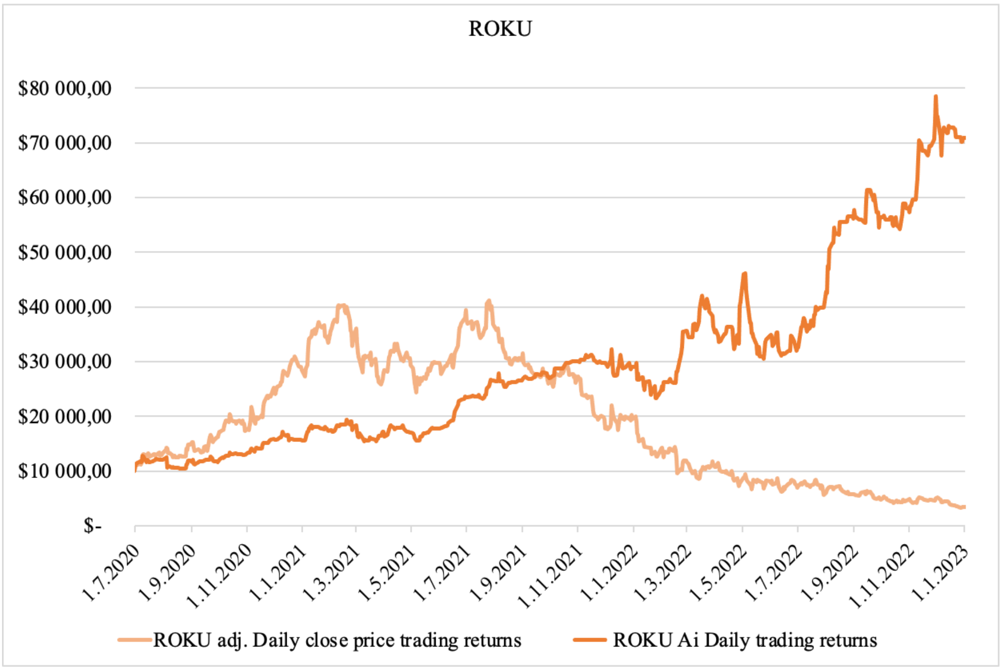

**Fig. 5. Ai trading results for ROKU compared with the Instrument Price Chart (Adj. Close Price) between 1st July 2020 and 1st January 2023**

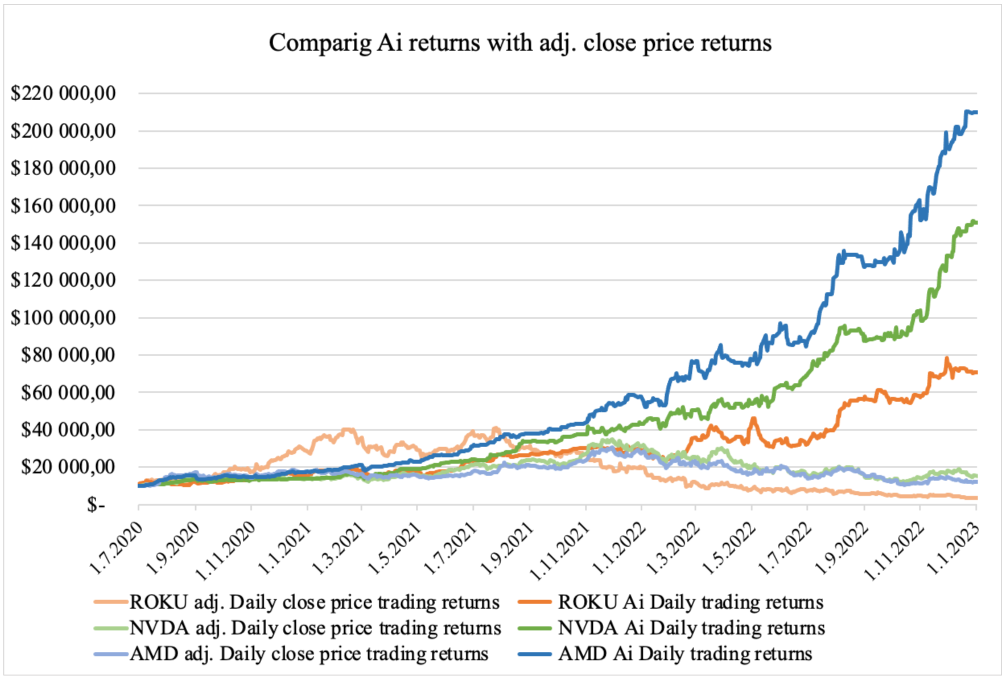

**Fig. 6. Ai trading results compared with the same instrument's daily market returns between 1st July 2020 and 1st January 2023**

By combining investments that have diverse levels of risk and return, we can create a portfolio that provides a balance between risk and return. The portfolio theory was developed by Harry Markowitz in the 1950s. The goal of portfolio theory is to create portfolios that maximize returns while minimizing risk, and it is based on the principle that diversification can reduce risk while the return of the portfolio is equal to the linear combination of the returns of the investment from the portfolio compiled. When constructing a portfolio, the aim is to diversify their risk by seeking assets with low or negative correlations. One method for achieving diversification is by examining the correlation between the assets. This is done by calculating the correlation coefficient, which measures the linear relationship between the variables. In this case, the results in Table 2 indicate that the returns of the chosen assets have a weak correlation with each other. This suggests that combining these assets in a portfolio would further reduce the risk and overall volatility.

**Tab. 2. Correlation coefficient between each asset**

| | **ROKU Ai returns** | **NVDA Ai returns** | **AMD Ai returns** |
|---|---:|---:|---:|
| **ROKU Ai returns** | 1 | | |
| **NVDA Ai returns** | 0.281290162 | 1 | |
| **AMD Ai returns** | 0.261986209 | 0.522698601 | 1 |

An equal-weighted portfolio consisting of three analyzed instruments, namely AMD, NVDA, and ROKU, was constructed, and its performance was evaluated over the period from July 1st, 2020, to January 1st, 2023. The outcome of this analysis revealed a return of 1348% for the specified time interval (Figure 7).

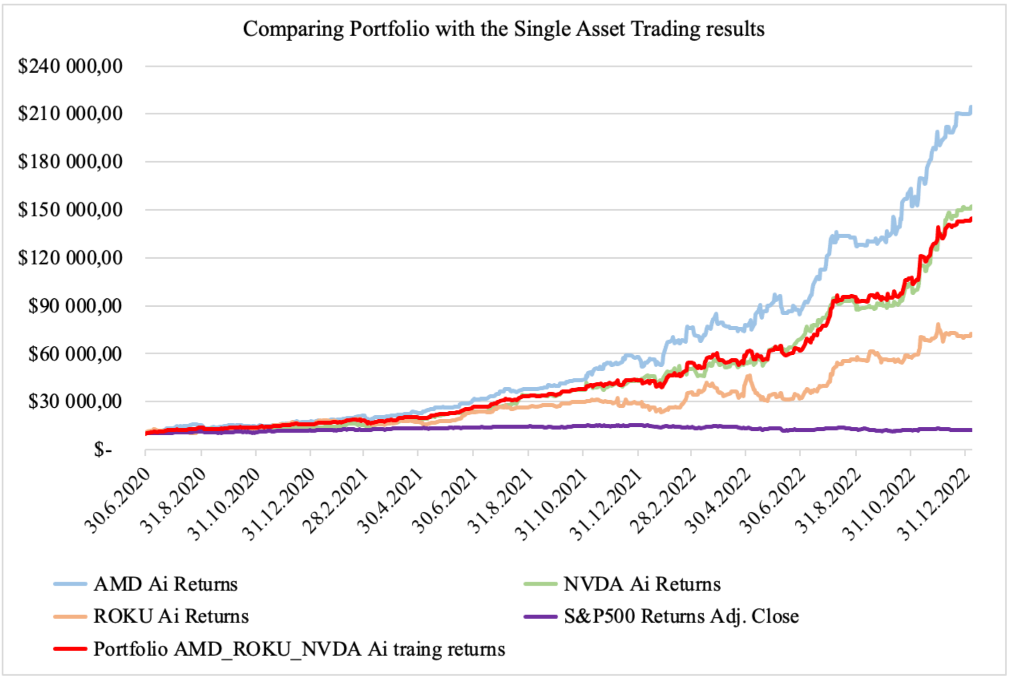

**Fig. 7. Ai trading results of the equal-weighted portfolio of AMD, NVDA, ROKU**

Regression analysis is a widely used statistical technique that is utilized to explore the relationship between an independent variable or multiple independent variables and a dependent variable. In this study, the authors employ regression analysis, using the market index daily adjusted close price returns as the independent variable, and the portfolio's returns as the dependent variable. The results are the following:

**Tab. 3. Regression results of market index returns, and the portfolio returns**

| **Regression Statistics** | |
|---|---:|
| Multiple R | 0.38691106 |
| R-Square | 0.14970016 |
| Adjusted R Square | 0.14835901 |
| Standard Error | 0.01710949 |
| Observations | 636 |

In Table 3., the R-squared value indicates that 14.97% of the variation in the dependent variable can be explained by the independent variable included in the model. The ANOVA significance F-value measures the overall significance of the regression model. The ANOVA F significance value in regression analysis indicates the overall significance of the regression model (Table 4).

**Tab. 4. ANOVA results of the regression on market index returns and the portfolio returns**

| | *df* | *SS* | *MD* | *F* | *Significance F* |
|---|---|---|---|---|---|
| Regression | 1 | 0.03267488 | 0.03267488 | 111.619341 | 3.83416E-24 |
| Residual | 634 | 0.18559394 | 0.00029273 | | |
| Total | 635 | 0.21826882 | | | |

The regression model is statistically significant, with both the intercept and independent variable having a significant impact on the dependent variable. However, the R-squared value suggests that the model only explains a small portion of the variation in the dependent variable, indicating that there may be other factors not accounted for in the model. The obtained intercept P-value of 2.1449E-09 is less than the significance level alpha, indicating that the constant parameter estimation i.e., the abnormal return of the investigated portfolio is statistically significant and gives the value of 0.4% daily return. However, the coefficient of the independent variable is only 0.60204, suggesting a weak correlation between the independent and dependent variables, that is the systematic risk of the portfolio under investigation is lower than the S&P 500 or by other words the sensitivity to the market volatility is less than unity (Table 4.).

**Tab. 5. Regression results of market index returns, and the portfolio returns**

| | Coefficient | Standard Error | t Stat | P-value | Lower 95% | Upper 95% | Lower 95.0% | Upper 95.0% |
|---|---|---|---|---|---|---|---|---|
| Alpha | 0.00412 | 0.00068 | 6.07449 | 2,1449E-09*** | 0.00279 | 0.00546 | 0.00279 | 0.00546 |
| Beta | 0.60204 | 0.05698 | 10.5650 | 3,8342E-24*** | 0.49014 | 0.71394 | 0.49014 | 0.71394 |

By using the returns of our simulated trading, the beta was calculated for each asset. The results, presented in Table 6., show that the beta for AMD is 0.589, NVDA is 0.662, and ROKU is 0.581. A beta less than 1.0 is typically considered less risky than the market, but it may also have lower returns if the regression alpha is not significant. In our case the Alphas are around 0.003 - 0.004 which indicates 0.3% - 0.4% abnormal reruns.

**Tab. 6. Beta for adj.close price daily returns, Beta and Alpha for the Ai trading daily returns**

| | AMD | NVDA | ROKU |
|---|---|---|---|
| Beta (adj.close price) | 1.873 | 2.097 | 1.908 |
| Beta Ai | 0.589 | 0.662 | 0.581 |
| Alphas | 0.004 | 0.004 | 0.003 |

## 5. CONCLUSIONS AND DISCUSSION

After conducting an analysis of the effectiveness of the authors' trading algorithm, the results show that it can generate abnormal returns and decrease risks of the individual securities. The Capital Asset Pricing Model (CAPM) and its alpha and beta values were used to measure performance against a market index, and the results shown in Table 5. and Figure 6. were encouraging. The Beta value for the algorithmic trading of NVDA, AMD and ROKU were close to zero, indicating that the instruments were independent from the market process, generating good diversification opportunities. This suggests that our algorithm was able to identify market trends accurately and generate profitable trades for AMD and ROKU, while mitigating risks for NVDA.

During the tested period, the abnormal returns of the trading algorithm were significant. The simulation showed that AMD had increased by 2046%, NVDA had grown by 1907% and ROKU had increased by 778%. These profits demonstrate the potential for AI based algorithmic trading to generate returns above market expectations. Additionally, our algorithm's returns were above the expected returns based on the CAPM model, further validating its effectiveness in generating profits while minimizing risks.

The results also suggest that the weak form of market efficiency can be argued as this trading strategy applies solely past market data by which our trading algorithm, employing a Support Vector Machine binary classification model, has really been able to generate statistically significant and exceptionally large abnormal returns.

It is essential to address the limitations of the developed algorithm to provide a comprehensive understanding of its applicability. Firstly, it should be acknowledged that the study primarily relies on simulation, raising uncertainty about its real-world effectiveness. The instantaneous nature of processes in simulation, such as prediction, bid placement, and trade execution, may not precisely mirror real-world scenarios, where even a brief delay could potentially impact outcomes. Furthermore, underlining the conditions for the algorithm's effective use is paramount. The algorithm demonstrates superior performance when intraday trading volumes and intraday volatility are high. In contrast, its efficacy diminishes under conditions of low trading activity and volatility.

Another crucial limitation of the developed algorithm relates to the scale of trading operations, specifically the size of contracts and the associated monetary value. The algorithm's effectiveness is constrained by the need to operate within small financial parameters, typically up to $2’000’000. This limitation arises from the potential for substantial trades to influence the market itself, leading to unintended consequences that may undermine the algorithm's predictive accuracy. In practice, larger investments could impact market dynamics, causing price shifts and altering trading conditions. Consequently, this algorithm is most suitable for traders and investors working with modest capital sizes. While its applicability to smaller-scale investments can still yield valuable insights and returns, its limitations become more pronounced when dealing with larger financial portfolios. Future research should address the algorithm's real-world feasibility and further refine its applicability in varying market conditions.

In conclusion, this paper offers a novel and significant contribution to the field of financial market analysis by displaying the profound impact of model selection, feature selection, and the utilization of high-frequency data for algorithmic day trading. The authors challenge the conventional Capital Asset Pricing Model (CAPM) and demonstrate the potential of the Support Vector Machines (SVM), for achieving superior prediction accuracy. Another result of the research is the exploration of the relationships among these factors. As a potential for future research, the presented method opens the possibility for further exploring stock price analysis in several directions. Advanced machine learning models can be explored, particularly deep learning algorithms, to assess their effectiveness in further enhancing prediction accuracy. Secondly, the development of hybrid models, integrating established financial theories such as CAPM with machine learning techniques, presents a valuable avenue for research to leverage the strengths of both approaches. Additionally, recurrence could be incorporated into these machine learning models, which would open the possibility of continuous real-time model improvement. Diversifying the scope of assets under examination, such as futures and options, can also provide a more comprehensive understanding of the applicability of machine learning across various financial instruments. Exploring these diverse research directions promises to deepen the knowledge of stock price dynamics understanding and broaden the practical applications of machine learning in finance.

## REFERENCES

Acharya, V. V., & Pedersen, L. H. (2005). Asset pricing with liquidity risk. *Journal of Financial Economics, 77*(2), 375-410. https://doi.org/10.1016/j.jfineco.2004.06.007

Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Stock price prediction using the ARIMA model. *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation,* (pp. 106-112). IEEE. https://doi.org/10.1109/UKSim.2014.67

Briola, A., Turiel, J., Marcaccioli, R., Cauderan, A., & Aste, T. (2021). *Deep reinforcement learning for active high frequency trading.* arXiv. https://doi.org/10.48550/arXiv.2101.07107

Ding, C., & Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology, 3*(2), 185–205. https://doi.org/10.1142/s0219720005001004

Srivastava, D., & Bhambhu, L. (2010). Data classification using support vector machine. *Journal of Theoretical and Applied Information Technology*, *12*(1), 1-7. Retrieved from http://www.jatit.org/volumes/research-papers/Vol12No1/1Vol12No1.pdf

Fama, E. F., & Laffer, A. B. (1971). Information and capital markets. *Journal of Business*, *44*(3), 289-298. http://dx.doi.org/10.1086/295379

Fama, E. F. (1991). Efficient capital markets: II. *The Journal of Finance*, *46*(5), 1575-1617. https://doi.org/10.1111/j.1540-6261.1991.tb04636.x

Fama, E. F., & French, K. R. (2004). The capital asset pricing model: Theory and evidence. *Journal of Economic Perspectives*, *18*(3), 25-46. https://doi.org/10.1257/0895330042162430

Grossman, S. J., & Stiglitz, J. E. (1980). On the impossibility of informationally efficient markets. *The American Economic Review*, *70*(3), 393-408. http://www.jstor.org/stable/1805228

Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2018). Stock price prediction using support vector regression on daily and up to the minute prices. *The Journal of Finance and Data Science*, *4*(3), 183-201. https://doi.org/10.1016/j.jfds.2018.04.003

Ph.-D. B. I. J., & Levy, K. N. (1989). The complexity of the stock market. *The Journal of Portfolio Management*, *16*(1), 19-27. https://ssrn.com/abstract=2447013

Jensen, M. C. (1968). The performance of mutual funds in the period 1945-1964. *The Journal of Finance*, *23*(2), 389-416. https://doi.org/10.1111/j.1540-6261.1968.tb00815.x

Ji, X., Wang, J., & Yan, Z. (2021). A stock price prediction method based on deep learning technology. *International Journal of Crowd Science*, *5*(1), 55-72. https://doi.org/10.1108/IJCS-05-2020-0012

Kohda, S., & Yoshida, K. (2022). Characteristics and forecast of high-frequency trading. *Transactions of the Japanese Society for Artificial Intelligence*, *37*(5), 1-9. https://doi.org/10.1527/tjsai.37-5_B-M44

Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1-2), 307-319. https://doi.org/10.1016/S0925-2312(03)00372-2

Lai, S., Wang, M., Zhao, S., & Arce, G. R. (2023). Predicting high-frequency stock movement with differential transformer neural network. *Electronics*, *12*(13), 2943. https://doi.org/10.3390/electronics12132943

Lintner, J. (1969). The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets: A reply. *The Review of Economics and Statistics*, *51*(2), 222–224. https://doi.org/10.2307/1926735

Lu, W., Li, J., Wang, J., & Oin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing & Applications 33*, 4741–4753. https://doi.org/10.1007/s00521-020-05532-z

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77–91. https://doi.org/10.2307/2975974

Merton, R. C. (1973). Theory of rational option pricing. *Bell Journal of Economics and Management Science*, *4*(1), 141-183. https://doi.org/10.2307/3003143

Mossin, J. (1966). Equilibrium in a capital asset market. *Econometrica, 34*(4), 768–783. https://doi.org/10.2307/1910098

Sapankevych, N. I., & Sankar, R. (2009). Time series prediction using support vector machines: A survey. *IEEE Computational Intelligence Magazine*, *4*(2), 24-38. https://doi.org/10.1109/MCI.2009.932254

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance*, *19*(3), 425-442. https://doi.org/10.1111/j.1540-6261.1964.tb02865.x

Treynor, J. L. (1965). How to rate management of investment funds. *Harvard Business Review*, *43*, 63-75. https://doi.org/10.1002/9781119196679.ch10

Snoek, J., Larochelle, H., & Adams, R. P. (2012). *Practical Bayesian Optimization of Machine Learning Algorithms*. arXiv. https://doi.org/10.48550/arXiv.1206.2944

Vapnik, V., & Cortes, C. (1995). Support-vector networks. *Machine Learning*, *20*, 273-297. https://doi.org/10.1007/BF00994018

Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A. (2020). Stock closing price prediction using machine learning techniques. *Procedia Computer Science*, 167, 599-606. https://doi.org/10.1016/j.procs.2020.03.326

Xu, X., Zhang, Y. (2023). Neural network predictions of the high-frequency CSI300 first distant futures trading volume. *Financial Markets and Portfolio Management*, *37*, 191-207. https://doi.org/10.1007/s11408-022-00421-y

Yu, P., Yan, X. (2020). Stock price prediction based on deep neural networks. *Neural Computing and Applications*, *32*, 1609-1628. https://doi.org/10.1007/s00521-019-04212-x

Zhang, Z., Khushi, M. (2020, July). Ga-mssr: Genetic algorithm maximizing sharpe and sterling ratio method for RoboTrading. *2020 International Joint Conference on Neural Networks (IJCNN)*(pp. 1-8). IEEE. https://doi.org/10.1109/IJCNN48605.2020.9206647