Spectrum of Operational Research

Journal homepage: www.sor-journal.org
ISSN: 3042-1470

Data-Driven Large-Cap US Stock Price Forecasting Using a Hybrid MCDM-Machine Learning Approach

Mehmet Özçalıcı[1,*], Nazlı Ersoy[2], Dragan Pamucar[3,4,5]

1 Department of International Trade and Logistics, Faculty of Economics and Administrative Sciences, Kilis 7 Aralık University, Kilis, Türkiye
2 Department of Business Administration, Faculty of Economics and Administrative Sciences, Osmaniye Korkut Ata University, Osmaniye, Türkiye
3 Faculty of Engineering, Dogus University, Istanbul, Türkiye
4 UNEC Applied Artificial Intelligence Research Center, Azerbaijan State University of Economics (UNEC), Baku, Azerbaijan
5 Department of Applied Mathematical Science, College of Science and Technology, Korea University, Sejong, Republic of Korea

ARTICLE INFO

Article history:
Received 7 October 2025
Received in revised form 19 November 2025
Accepted 16 January 2026
Available online 26 January 2026

Keywords:
Stock price prediction; Neural network; Transformer; WENSLO; CRITIC; MEREC; SECA; MCDM; Decision making

ABSTRACT

This study investigates the enhancement of stock price forecasting for the large-cap sector of the U.S. market through the integration of objective weights derived from multi-criteria decision-making (MCDM) methods. Due to the inherent complexities and volatility of stock markets, traditional forecasting approaches often struggle to provide accurate predictions. In this research, MCDM techniques such as EW, SD, MEREC, SECA, CRITIC, CILOS, and WENSLO are applied to assign appropriate weights to various input features, thereby improving the predictive capabilities of stock price models. The proposed methodology utilizes historical stock price data alongside objectively determined weights to train advanced machine learning models, including transformer neural networks. The findings of this study demonstrate that the application of MCDM weighting techniques to input features significantly enhances the forecasting performance of the transformer neural network model. This research not only contributes to the field of stock price prediction but also offers a framework for applying objective feature-weighting methods in various financial forecasting contexts.

1. Introduction

The stock market functions as a mechanism that facilitates the trading of stocks among investors, and it constitutes a critical component of a country's economy, as it serves as an essential means for companies to generate funds [1]. In recent years, stock trading has garnered significant attention, primarily due to advancements in technology. Investors are increasingly seeking strategies and tools that can enhance profitability while minimizing risk [2]. In this context, the prediction of stock prices has drawn interest from both investors and researchers because of its substantial potential value [3]. Nevertheless, examining stock market trends and price dynamics poses significant challenges due to the market's naturally fluctuating, unpredictable, evolving, parameter-free, noisy, and chaotic nature

* *Corresponding author.*
E-mail address: mozcalici@kilis.edu.tr

https://doi.org/10.31181/sor202771

[4]. Economic analysts and stock traders have employed a range of models to forecast stock performance, such as the quantile autoregression model (QAR) [5], hidden Markov model [6], deep neural networks (DNN) [7], and recurrent neural networks (RNN) [8-10].

Within the realm of powerful prediction models, deep learning (DL) techniques have gained popularity due to the increase in available datasets and the advancement of computational power [11]. Although there have been some works on deep learning for stock prediction, there are many modern models that disregard the importance of data preprocessing, instead choosing to rely on data pattern discovery by models directly from raw data. This is, although possible through deep learning, often fraught with difficulties due to the non-linear properties of financial data [12]. To improve the precision of stock trend forecasting and investigate the use of transformer models in this area, this study integrates multi-criteria decision-making (MCDM) weighting techniques into the transformer neural network model.

In numerous MCDM methods, the weights assigned to criteria during the aggregation process are essential for evaluating the overall preferences of alternatives. Given the diversity of aggregation rules, various weighting techniques have been developed for application in different MCDM frameworks [13]. There are two main types of weighting approaches: objective and subjective. In subjective methods, which use the judgment of the decision-maker, the results or rankings obtained from the analysis may be impacted by the level of expertise of the decision-maker in the area [14]. Conversely, objective methods use mathematical formulas to establish criterion weights and are commonly used because they do not depend on specialized knowledge of the specific issue, thus improving the automation of decision support systems [15].

In the literature, we observe that the weighting techniques are often used in conjunction with other MCDM methods [16-18]. However, this study examines a different approach by integrating objective techniques into the transformer neural network model to investigate the resulting outcomes. While historical price data serves as input for the model, the weights of these data are determined using objective MCDM techniques, and we evaluate whether there is an improvement in performance through statistical tests. In the dataset divided into training and test sets, each input set used during the training phase was weighted separately according to the techniques of equal weight, standard deviation weight, CILOS, CRITIC, MEREC, SECA, and WENSLO. Moreover, the case without weighting was also considered. Each weighting method was applied to the data set in the process of training the model. The model was trained separately for each weighting method.

In recent years, the increased use of machine learning methods, particularly transformer neural networks, in stock price forecasting has made research in this field even more significant. However, alongside these advancements, the neglect of feature weighting can limit the potential performance of models. This study aims to contribute to the existing literature by highlighting the role of objective weighting in stock price forecasting through the application of MCDM techniques. The findings demonstrate the importance of objective weighting in enhancing the accuracy of stock price predictions, offering a new perspective in the field of financial analytics.

The contributions of this study can be summarized as follows:

i. This research leverages the Transformer neural network, a recently developed and advanced model, specifically applied for stock price prediction.
ii. To enhance the predictive performance of this model, it is integrated with multi-criteria decision-making (MCDM) techniques that assign objective weights to relevant criteria, providing a novel approach to feature weighting in financial forecasting.
iii. The study explores model performance across test sets of varying lengths, assessing robustness and adaptability under different forecasting conditions.

iv. Statistical tests are conducted to verify and substantiate any observed improvements in performance, ensuring that the enhancements are both significant and reliable.

This approach aims to establish a more accurate and rigorous method for stock price prediction by combining state-of-the-art neural networks with objective weighting techniques.

The remainder of the study is organized as follows: The second section summarizes studies on AI-based stock price prediction. The third section provides detailed information about the MCDM criteria weighting methods used. The fourth section explains the transformer model. Sections five and six are dedicated to methodology and application, respectively. The final section is devoted to the conclusion.

2. Literature review

AI simulates human intelligence and surpasses humans significantly in terms of computing speed, capacity, and accuracy. It facilitates the continuous expansion of human cognition into both micro and macro levels [19]. For addressing the needs that have arisen out of the present scenario of cybersecurity threats, different artificial intelligence solutions can be leveraged for carrying out intrusion detection and prevention. In this respect, different solutions include machine learning (ML), natural language processing (NLP), deep learning (DL), knowledge representation and reasoning (KRR), and rule-based and knowledge-based expert systems [20].

Advancements in the field of DL have revolutionized the application of Deep Neural Networks (DNNs). DNNs consist of multiple hidden layers, with each layer extracting complex representations from the input data and passing them to the next layer [21]. The major drawback of deep neural networks is the difficulty of handling long-term dependencies. Although derivative architectures like recurrent neural networks and long short-term memory networks have alleviated this issue to some extent, these networks still face problems of performance while handling long sequences [22]. However, there has been significant progress that has been achieved through the latest breakthroughs in deep learning techniques, especially in the fields of natural language processing and image processing. In this regard, the transformer neural network model has been a transformative idea for dealing with complex data [23].

The stock market forecast is primarily concerned with the forecast for stock prices and stock movements to evaluate the forecasted trends in the market and the risks involved in the trends. The stock market trends are the direction in which the stock market is moving, either upwards or downwards [24]. On the other hand, stock market predictions present a challenging problem due to the inherent uncertainties associated with market movements [25]. Forecasting models within financial markets have been traditionally divided into two approaches. These include fundamental analysis and technical analysis [26]. In this section, in addition to these two categories, we have included studies on autoregressive analysis related to stock price prediction.

2.1. Fundamental analysis

Fundamental analysis assesses stock prices by examining their intrinsic value, or fair value, whereas technical analysis focuses solely on charts and trends [27]. It combines qualitative market information (firm descriptions) with quantitative market information (stock prices) [28]. Financial analysts make use of textual information in financial reports to guide investment decisions through market sentiment analysis [29].

[30] define sentiment analysis (SA) as a method in NLP aimed at detecting sentiments, opinions, and emotions embedded in unstructured data. [31] utilized two deep learning models (Deep Belief Networks and Recurrent Neural Networks) for sentiment analysis aimed at forecasting the Japanese stock market. [32] tested the efficiency of transformer-based architectures for a sentiment analysis

problem in the Turkish language on the Twitter platform. Likewise, [33] have proposed a stock market predictive system, where the inputs to the model are a combination of news and financial data, and the model is a long short-term memory (LSTM) neural network, specifically for the stock exchange in Hong Kong.

2.2. Technical analysis

Technical analysis centers on examining historical stock prices to forecast future values, emphasizing the directional movement of prices [26]. The first difference is regarding the rate at which new information is integrated into the stock price formation process. Technical analysts begin with the premise that stock prices are driven by news with market-related factors, including the psychological aspects, being constant. Consequently, this leads to a level of stability that makes it possible for prices to exhibit persistent patterns that are recurrent and predictable [34]. [35] developed an extensive methodology for predicting stock indices, which integrates iterative dropout testing and optimization of batch sizes. The newly developed hybrid neural network model was named the Gated Recurrent Unit (GRU) Transformer. This model was created by [11]. In another research, [36] used the Transformer model for predicting the closing stock prices of companies underlying the China Securities 50 Index.

2.3. Autoregressive

Autoregressive (AR) models, which are regression models applied to lagged series derived from the original time series, represent the output as a future data point. In multiple linear regression, the output is derived from a linear combination of several input variables [37]. The studies using autoregressive inputs for stock price forecasting can be listed as follows. To overcome the problem of non-stationarity in stock market data, a fuzzy stochastic autoregressive model using triangular fuzzy number data was proposed by [38]. To forecast the stocks in the Dhaka Stock Exchange (DSE) market, ARIMA models combined with artificial neural networks (ANN) have been used by [39].

3. Weighting methods

This section summarizes the objective weighting methods used and presents the associated procedures and calculation steps in detail in Table 1.

Table 1
The algorithms of the MCDM methods

Method/Reference	Steps	Equation	
SD [40]	1: Calculation of criteria weights	$w_i = \frac{\sigma_j}{\sum_{k=1}^{m} \sigma_k}$ j=1,2,…,m	(1)
MEREC [41]	1: Normalize the decision matrix	$n_{ij}^{x} = \begin{cases} \frac{\min_k x_{kj}}{x_{ij}} & for\ benefit \\ \frac{x_{ij}}{\max_k x_{kj}} & for\ cost \end{cases}$	(2)
	2: Calculate the overall performance of the options	$S_i = \ln(1 + (\frac{1}{m}\sum_j \lceil \ln(n_{ij}^{x}) \rceil))$ m is the number of criteria	(3)
	3: Calculate the performance of the alternatives by excluding each criterion	$S'_{ij} = \ln(1 + (\frac{1}{m}\sum_{k,k\neq j} \lceil \ln(n_{ik}^{x}) \rceil))$	(4)
	4: Calculate the total of absolute deviations	$E_j = \sum_i \lvert S'_{ij} - S_i \rvert$ E_j represents the impact of excluding the jth criterion.	(5)

Table 1
Continued

Method/Reference	Steps	Equation
MEREC [41]	5: Calculation of criteria weights	$w_j = \frac{E_j}{\sum_k E_k}$ (6) w_j signifies the weight assigned to the jth criterion.
SECA [42]	1: Normalize the decision matrix	$x_{ij}^N = \begin{cases} \frac{x_{ij}}{max_k x_{kj}} & if\ j\ \in BC \\ \frac{min_k x_{kj}}{x_{ij}} & if\ j\ \in NC \end{cases}$ (7) x_{ij}^N is the normalized value of alternative i in terms of criterion j, BC and NC refer to useful and non-useful criterion sets, respectively
	2: Determine the vector of each criteria	$\pi_j = \sum_{l=1}^{m} (1 - r_{jl})$ (8) rjl shows the correlation of normalized decision matrix columns, πj represents the level of conflict between criterion j and the other criteria.
	3: tandard deviation values are calculated	$\sigma_j = \sqrt{\frac{x_{ij-}^N \overline{x_{ij}^N}}{m}}$ (9) σ_j is the standard deviation value of criterion j
	4: Normalize the σj and πj as the reference points	$\sigma_j^N = \frac{\sigma_j}{\sum_{l=1}^{m} \sigma_1}$ (10) $\pi_j^N = \frac{\pi_j}{\sum_{l=1}^{m} \pi_1}$ (11) σ_j^N is the normalized standard deviation, x_j^N is the normalized conflict degree
	5: The multi-objective nonlinear model is solved	$\max S_i = \sum_{j=1}^{m} w_j\, x_{ij}^N, \forall_i \in \{1,2,\dots,n\}$ $\min \lambda_b = \sum_{j=1}^{m} (w_j - \sigma_j^N)^2$ $\min \lambda_c = \sum_{j=1}^{m} (w_j - \pi_j^N)^2$ (12) $s.t. \sum_{j=1}^{m} w_j = 1$ $w_j \leq 1, \quad \forall_j \in \{1,2,\dots,m\}$ $w_j \geq \varepsilon, \quad \forall_j \in \{1,2,\dots,m\}$ S_i is the performance value of alternative, wj refers to the weight of the criterion, ε refers to the criterion weight lower bound parameter ($\varepsilon = 0{,}001$).
	6: The multi-objective optimization model used to optimize the model in Step 5 was transformed	$max\ Z = \lambda_a - \beta \left(\sum_{j=1}^{n} \left(w_j - \sigma_j^N\right)^2 + \sum_{j=1}^{n} \left(w_j - \pi_j^N\right)^2 \right)$ $\lambda_a \leq \sum_{i=1}^{m} w_j x_{ij}^N, \quad \forall j \in \{1,2,\dots,n\}$ $\sum_{j=1}^{n} w_j = 1$ (13) $w_j \leq 1, \quad \forall j \in \{1,2,\dots,n\}$ $w_j \geq \epsilon, \quad \forall j \in \{1,2,\dots,n\}$

Table 1
Continued

Method/Reference	Steps	Equation
	1: Normalize the decision matrix	$r_{ij} = \frac{x_{ij} - x_j^{min}}{x_j^{max} - x_j^{min}}\ benefit$ (14) $r_{ij} = \frac{x_j^{max} - x_{ij}}{x_j^{max} - x_j^{min}}\ cost$ (15) $i = 1,\ldots,m;\ j = 1,\ldots,n$
CRITIC [43]	2: Determine the linear correlation matrix	$p_{jk} = \frac{\sum_{i=1}^{m}(r_{ij} - \bar{r}_j)(r_{ik} - \bar{r}_k)}{\sqrt{\sum_{i=1}^{m}(r_{ij} - \bar{r}_j)^2 \sum_{i=1}^{m}(r_{ik} - \bar{r}_k)^2}}$ (16) $j, k = 1,\ldots,n$ p_{jk} represents the correlation coefficient between the rj and rk vectors.
	3: Calculate the key indicator and weight of criteria	$c_j = \sigma_j \sum_{k=1}^{n}(1 - p_{jk})$ (17) $w_j = c_j / \sum_{k=1}^{n} ck \quad j = 1,\ldots,n$ (18) cj is the information given by j th indicator and pjk is the linear correlation between indicators j and k. wj is the weight of j th indicator.
	1: Transformation of cost criteria	$\bar{r}_{ij} = \frac{min_i r_{ij}}{r_{ij}}$ (19)
	2: Normalizing the transformation matrix	$x_j = \frac{x_{ij}}{\sum_{i=1}^{n} x_{ij}}$ (20)
	3: Obtaining Square Matrix A	A= $\|\|a_{ij}\|\|$, $a_{ii} = x_i, a_{ij=}x_{kij},)i, j = 1,2,\ldots,m)$ (21) m is the number of criteria
CILOS [44]	4: Obtaining the relative loss matrix P	$p_{ij} = \frac{a_{ii} - a_{ij}}{a_{ii}}, (p_{ii} = 0;\ i, j\ = 1,2,\ldots,m)$ (22) p_{ij} indicates the relative loss associated with the jth criterion.
	5: Determination of the weight system matrix F	$F = \begin{pmatrix} -\sum_{i=1}^{m} p_{i1} & p_{12} & \cdots & p_{1m} \\ p_{21} & -\sum_{i=1}^{m} p_{i2} & & p_{2m} \\ & \cdots & & \\ p_{m1} & p_{m2} & \cdots & -\sum_{i=1}^{m} p_{im} \end{pmatrix}$ (23)
	6: Solving the linear equation system	$Fq^T = 0$ (24)
	1: Normalization of input data	$z_{ij} = \frac{\zeta_{ij}}{\sum_{i=1}^{m} \zeta_{ij}}, \forall_j \in [1,2,\ldots,n]$ (25) m and n represents number of alternatives and criteria, ζij is the estimated value for the ith alternative in relation to the jth criterion.
WENSLO [45]	2: Calculation of criterion class interval	$\begin{bmatrix} A \\ A_1 \\ A_2 \\ \vdots \\ A_i \end{bmatrix} = \left(\begin{bmatrix} C_1 \\ z_{11} \\ z_{21} \\ \vdots \\ z_{i1} \end{bmatrix}, \begin{bmatrix} C_2 \\ z_{12} \\ z_{22} \\ \vdots \\ z_{i2} \end{bmatrix}, \ldots, \begin{bmatrix} C_j \\ z_{1j} \\ z_{2j} \\ \vdots \\ z_{ij} \end{bmatrix} \right)$ The size of the jth criterion class interval Δ_{zj} is calculated by Sturges' rule as; $\Delta_{z_j} = \frac{\max_{i=1,2,\ldots,m} z_{ij} - \min_{i=1,2,\ldots,m} z_{ij}}{1 + 3.222 * log(m)}, \forall_j \in [1,2,\ldots,n]$ (26) Zij represents the element of the normalized decision matrix
	3: Calculation of the criterion slope	$\tan \phi_j = \frac{\sum_{i=1}^{m} z_{ij}}{(m-1) * \Delta z_j} \forall_j \in [1,2,\ldots,n]$ (27)

Table 1
Continued

Method/Reference	Steps	Equation
WENSLO [45]	4: Determination of the criterion envelope	$E_j = \sum_{i=1}^{m-1} \sqrt{(z_i + 1, j - z_{i,j})^2 + \Delta z_j^2}$ (28)
	5: Define the envelope–slope ratio	$q_j = \frac{E_j}{tan\phi_j} \ \forall_j \in [1,2,\ldots,n]$ (29)
	6: Calculation of criteria weights	$w_j = \frac{q_j}{\sum_{j=1}^{n} q_j} \ \forall_j \in [1,2,\ldots,n]$ (30)

4. Transformer neural network model

The Transformer model was developed to handle sequential data with greater granularity [11]. Initially introduced by [23], it has since inspired numerous effective implementations in areas such as language analysis and visual recognition [46]. The Transformer model consists of multiple layers of self-attention and feed-forward networks in every stage in both the encoder and the decoder [23], shown in the left and right side of Figure 1, respectively. The basic elements of the Transformer model are the input layer, the encoder, the attention mechanism, and the decoder [47].

Input-Output: The encoder initially receives the input, which passes through a self-attention layer; this layer allows the encoder to look at other words in the input it receives. This is followed by a fully connected feedforward neural network which takes the form of a simple hierarchical structure [48].

One challenge of Transformer's nonsequential data processing is handling historical information. To tackle this, positional encoding [49] is introduced to supply context regarding the relative distances between tokens and the current timestamp [47]. Positional information is encoded using sine and cosine functions across various frequencies [23].

$$PE_{(pos,2i)} = \sin\left(pos/10000^{2i/d_{model}}\right)$$
$$PE_{(pos,2i+1)} = \cos\left(pos/10000^{2i/d_{model}}\right)$$

where pos and i represent the position and dimension. Each aspect of the positional encoding is linked to a sinusoidal function. The wavelengths increase geometrically from 2π to 10000* 2π. This function was chosen under the assumption that it would enhance the model's capacity to focus on relative positions, as PE_{pos+k} can be represented as a linear function of PE_{pos} for any fixed offset k.

Encoder-Decoder: The encoder condenses essential information from the input sequence into a fixed-size vector, which is subsequently transformed into the output by the decoder [50]. The encoder (depicted in Figure 1, left block) is built from N identical layers, each consisting of two components: a multi-head self-attention mechanism followed by a position-based feedforward network. The decoder, illustrated in Figure 1 (right block), shares much of its structure with the encoder, but it comprises two layers of multi-head self-attention instead of one [29]. As in the encoder, all sub-layers in the decoder consist of residual connections followed by layer normalization. However, modifications are made to the decoder's self-attention mechanism to limit attention only to previous locations, thus concealing connections to later locations. The masking, together with the one position shift in the output embeddings, ensures that the prediction at position i is dependent only on the preceding outputs [23].

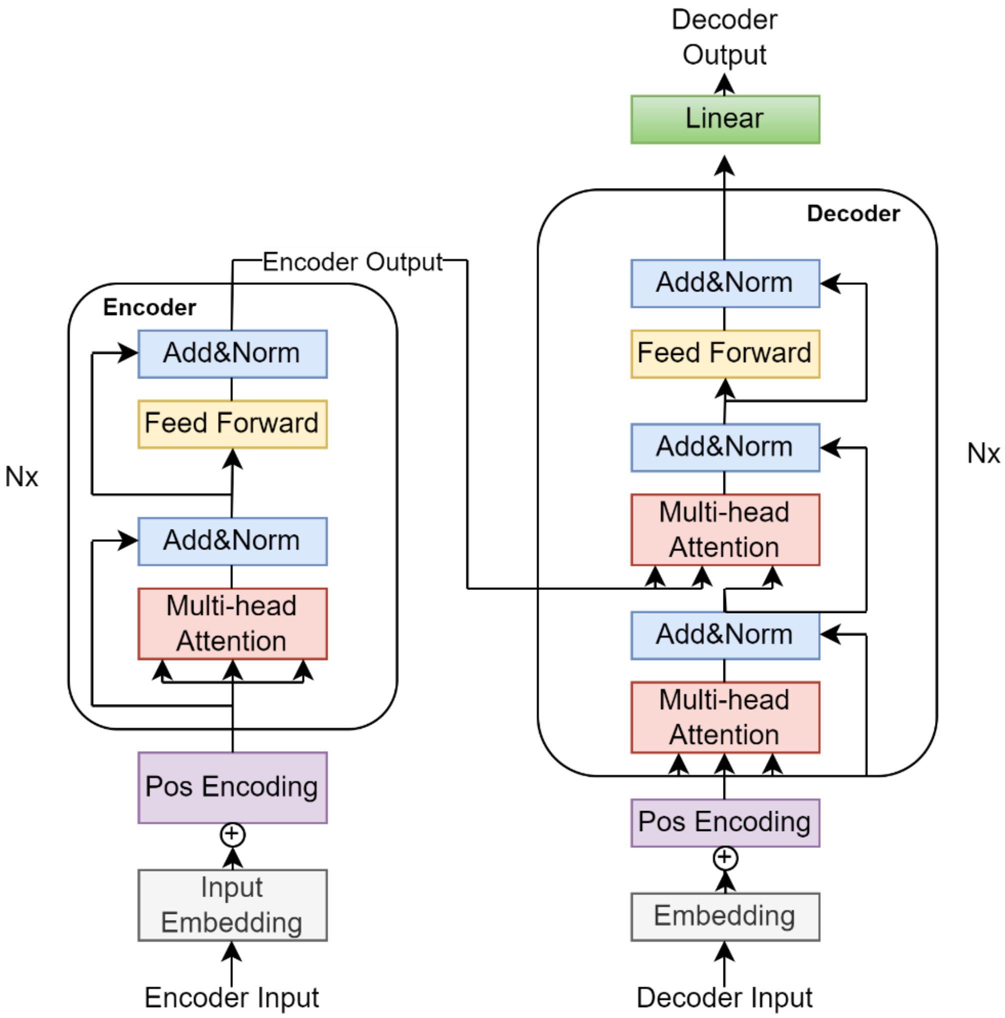


Fig. 1. The general architecture of a Transformer model

Self-Attention: In Transformer architecture, the self-attention layer addresses the vanishing gradient issue by enabling the model to retrieve information from any prior location within the sequence, thereby enhancing its ability to learn from long-range dependencies [35]. [23] describe the self-attention mechanism as follows:

$$Attention\ (Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (1)$$

where $Q\varepsilon R^{Txd}$, $K\varepsilon R^{Txd}$ *and* $V\varepsilon R^{Txd}$ denote the query, key, and value matrices, respectively, generated from three distinct linear layers using the same input.

Multi-head Attention: Rather than executing a single attention operation with queries, keys, and values in d model dimensions, it has been found beneficial to project the queries, keys, and values h times into dk, dk, and dv dimensions, respectively, using distinct learned linear transformations. The resulting transformation enables parallel application of the attention operation to obtain output values with dimensions of dv [23]. The mathematical expression for multi-head attention is provided as follows [23]:

$$MultiHead(Q, K, V)cONTACT(head_1, .., head_h)W^o$$

where $head_i = Attention\ (QW_i^Q, KW_i^K, VW_i^V)$

where the projections are represented by parameter matrices $W_i^Q \in R^{d_{model} x d_k}, W_i^K \in R^{d_{model} x d_k}, W_i^V \in R^{d_{model} x d_k}$ and $W^o \in R^{hd_v x d_{model}}$.

5. Methodology

5.1. Proposed method

Figure 2 outlines the study framework, which examines the impact of objective MCDM weighting techniques on stock price forecasting performance. After computing input–output values, inputs are standardized using Z-score normalization and the dataset is split into training and test sets. During training, inputs are weighted using eight techniques: no weight, equal weight, standard deviation, CILOS, CRITIC, MEREC, SECA, and WENSLO, and a separate model is trained for each.

The stock prices are predicted using the Transformer model, in which the parameters of the model are trained using two stages without weighting. After this process, the model is further trained using each weighting method, with the respective weights being used for the test data. Normalization is then reversed, and performance is evaluated using RMSE, MAPE, and Hit Rate. The ANOVA test is used to determine whether there is a significant difference in performance due to feature weighting.

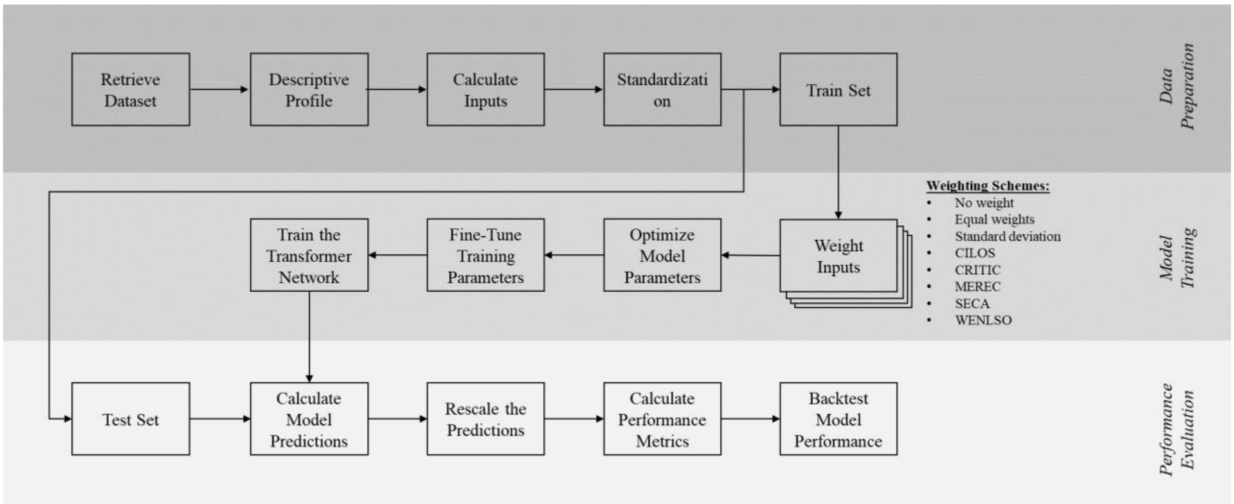


Fig. 2. Outline of the study

6. Analysis

6.1. Dataset Description

As of September 2024, the six leading large-cap firms in the US market are Apple, Microsoft, NVIDIA, Alphabet, Amazon, and Meta. Since Alphabet and Meta began trading more recently, this study focuses on large-cap stocks with continuous data availability since January 1, 2000: Apple (AAPL), Microsoft (MSFT), NVIDIA (NVDA), and Amazon (AMZN). Table 2 summarizes the descriptive statistics of the dataset, which is divided into a training set (January 3, 2000–December 30, 2022; 5,787 observations) and a testing set (January 3, 2023–September 18, 2024; 430 observations). The data from the training sets show the presence of variability for the stock prices. For example, the price variability for AAPL is wide, while the stock with the highest standard deviation is MSFT. Skewness and kurtosis values suggest right-skewed distributions and pronounced peakedness, particularly for NVDA, AAPL, and MSFT, implying the presence of extreme values. The test data reflects market characteristics which are more recent, as suggested by the high average prices for AAPL and MSFT according to their growth patterns. Compared to the training data, there is a decrease in standard deviation, which is an indication of reduced market volatility. The skewness is close to zero, and the kurtosis is still moderate.

Table 2

Descriptive profile of the dataset

		Min	Max	mean	%25	%50	%75	std	Skewness	Kurtosis	JB Stat	P	ADF test	p
Training Set	APPL	0.23	182.01	29.96	1.87	12.76	35.46	43.17	1.95	5.84	5620.99	0.001	1.23	0.94
	AMZN	0.30	186.57	34.30	2.20	9.29	44.52	49.42	1.59	4.31	2863.64	0.001	-0.23	0.57
	MSFT	15.15	343.11	68.86	26.95	31.54	65.48	75.41	1.98	5.80	5687.07	0.001	1.11	0.93
	NVDIA	0.06	33.38	3.00	0.29	0.45	2.71	5.73	2.64	9.85	18041.58	0.001	-0.32	0.53
Testing Set	APPL	125.02	234.82	182.43	170.03	181.17	192.58	22.22	0.16	3.06	1.99	0.340	1.58	0.97
	AMZN	83.12	200.00	144.76	123.43	142.97	176.76	32.02	-0.10	1.75	28.51	0.000	1.44	0.96
	MSFT	222.31	467.56	357.41	318.95	361.95	413.52	61.30	-0.31	2.05	22.77	0.000	1.58	0.97
	NVDIA	14.27	135.58	61.70	39.33	47.10	88.96	34.28	0.64	2.12	43.71	0.000	1.38	0.96
Starting date of training set :				03 – January – 2000		Ending date of training set :			30 – December – 2022			number of Obs : 5787		
Starting date of testing set :				03 – January – 2023		Ending date of testing set :			18 – September – 2024			number of Obs : 430		

Figure 3 presents the temporal fluctuations in the prices of the stocks examined in this study. A marked increase in stock prices is particularly evident from 2020 onwards. In contrast, prior to this period, the prices exhibited relative stability. The data for the years 2023 and 2024 are employed for testing purposes.

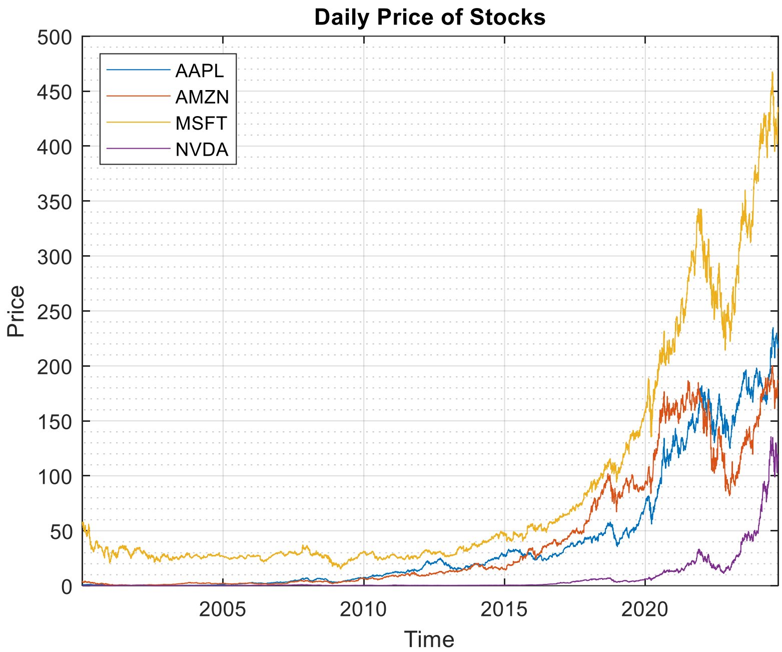


Fig. 3. Daily prices for stocks (last two years are in testing set)

6.2. Paramter optimization

The parameter determination process for the primary prediction model, the Transformer neural network, is conducted in two stages. In the first stage, the parameters for the model are defined. At the next stage, the training parameters receive a fine-tuning. Throughout the two processes for the parameters, the no-weighting method is used, ensuring that all parameters receive equal consideration without assigning any weights to any particular factor.

6.3. Determining model parameters

Critical parameters that affect the performance of the Transformer neural network model in the optimization of parameters include the maximum position, number of attention heads, and number of key channels. The search ranges for these parameters are outlined in Table 3. For each parameter, four distinct values were tested, resulting in a total of $4^3 = 64$ experiments. The corresponding RMSE values were calculated for these experiments. During the experiments performed with model parameters, the training parameters were kept constant at the following values: mini batch size = 64, learning rate = 0.001, solver = Adam, shuffle = once, and gradient threshold = 1.

Analysis of Variance was performed to determine the significance of the parameters on the value of the dependent variable, RMSE. The parameters were taken as the independent variables. The summary results are reported in Table 3, with detailed outcomes provided in the appendix. Analysis of the data obtained from the experiment has shown that there are no significant differences between the parameters concerning the RMSE. The lowest value of the RMSE is 0.1224, and it is obtained at the maximum position of 256, the head count of 8, and 24 key channels.

Table 3
Search range and ANOVA results for modeling parameters

		ANOVA Summary	
	Search Range	F value	p
Maximum Position	[32 64 128 256]	1.82	0.1528
Number of Heads	[3 4 8 16]	0.92	0.4345
Number of Key Channel	[12 24 32 64]	1.44	0.2396

6.4. Determining training parametres

The fine-tuning process has been performed for the selected model parameters and training parameters. The training parameters include mini batch size, learning rate, solver, shuffle, and gradient threshold. The search range values related to these parameters are presented in Table 4. A total of 432 experiments were conducted, calculated as (4^2)(3^3). RMSE values were recorded for each component. The ANOVA analysis was conducted using the RMSE results as the dependent variable while the components acted as the independent variables. The results are shown in Table 4.

Based on the ANOVA result, it can be seen that the gradient threshold parameter is not statistically significant to RMSE, while the other parameters have significance to RMSE. The result is shown in the appendix. The component that yielded the lowest RMSE value was found to be: mini batch size = 32, learning rate = 0.01, solver = msprop, shuffle = every epoch, and gradient threshold = 10.

Table 4
Search range and ANOVA results for training parameters

		ANOVA Summary	
	Search Range	F value	p
Mini batch size	[32 64 128 256]	7.81	4.36e-05
Learning rate	[1e-02 1e-03 1e-04 1e-05]	17.37	1.17e-10
Solver	[sdgm msprop adam]	8.47	0.0002
Shuffle	[once never every-epoch]	202.96	9.33e-63
Gradient Threshold	[1 4 10]	0.80	0.4521

The importance of parameter choice in improving predictive accuracy is clear. Although the effect of the gradient threshold on the value of RMSE is small, the other parameters, which include the size of the mini-batch, learning rate, solver, and shuffling of data, are found to be the key determinants of accuracy in the developed model. Figure 4 illustrates the relationship between performance metrics during the optimization of both model and training parameters. The results show that model parameters yield performance metrics concentrated within a narrow range, indicating a stable but limited impact on prediction accuracy. In contrast, training parameters produce a much wider dispersion of metrics, consistent with the hypothesis test results. These findings suggest that the model parameters have a relatively stable effect on performance, while the training parameters have a more critical effect on predictive performance. It is, therefore, necessary to carefully consider and optimize the training parameters in order to have a better performance in terms of forecast accuracy.

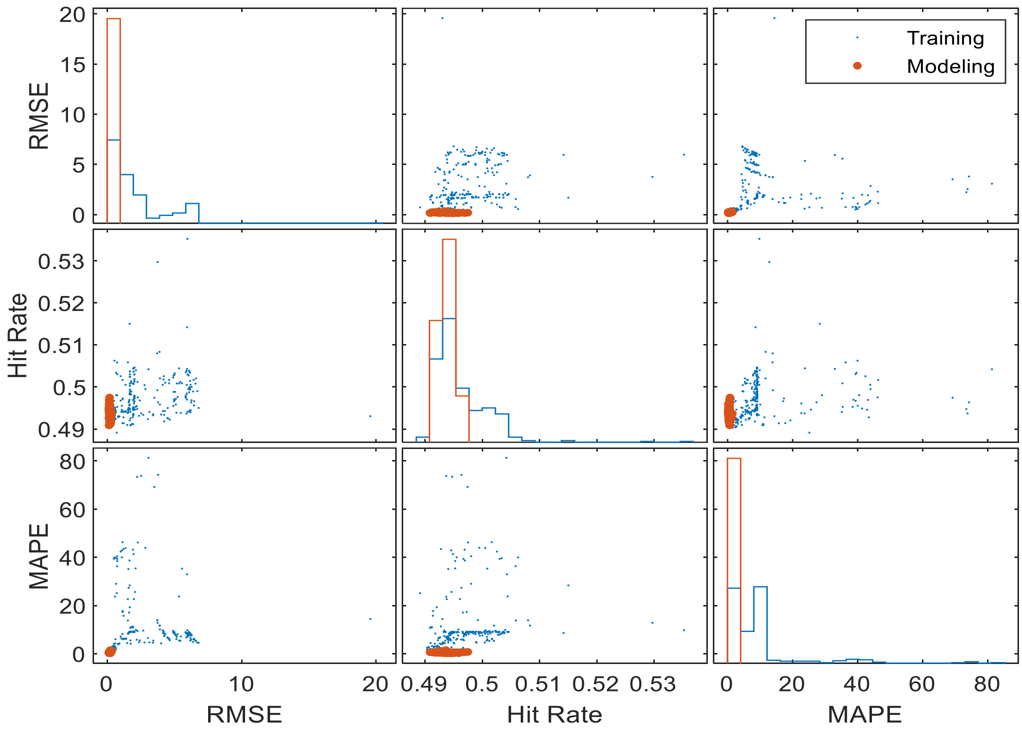


Fig. 4. Relationship between performance evaluation metrics in modeling and training selection phases

6.5. Determining criteria weights

The weights assigned by different MCDM tools for the dataset containing the stock prices for the preceding 30 days are presented in Figure 5. The days are listed from the most recent day (Day 1) to the least recent day (Day 30) for each stock price. The following discussion will critically analyze the weight assignments by different tools for their possible applications in stock price predictions.

Equal Weighting: Assigns identical weights (0.0333) to all 30 days, assuming equal importance across time. Although simple and neutral, it ignores temporal dynamics and may be suboptimal if recent prices have higher predictive relevance.

Standard Deviation Weighting: Weights increase slightly from Day 1 (0.03314) to Day 30 (0.03351), reflecting variability in the data and a mild emphasis on recent observations. However, the small differences may be insufficient to capture strong time-based trends.

CILOS: Allocates decreasing weights from Day 1 to Day 30, prioritizing recent data, with notable spikes on Days 24 and 25. This can enhance short-term forecasting but may underrepresent longer-term trends.

CRITIC: The algorithm gives relatively greater weights to the last days (ranging from 0.0479 on Day 1 to 0.0485 on Day 30) based on contrast intensity and inter-temporal correlation. Although useful in case of correlated data, its relatively uniform distribution might weaken the importance of important times.

MEREC: Produces an almost uniform weight distribution, ranging from 0.03313 (Day 1) to 0.03362 (Day 30). This ensures stability but may underperform when temporal patterns are pronounced.

SECA: Displays a downward weighting trend, with values decreasing from 0.0407 (Day 1) to 0.0409 (Day30), emphasizing earlier data and long-term trends, which may be less suitable for volatile markets.

WENSLO: Generates a highly skewed distribution, assigning very small weights to most days (e.g., 0.0000931 on Day 1) and an exceptionally large weight to Day 25 (0.7937). While effective for capturing sharp market movements, this concentration may increase the risk of overfitting.

The differences in the weightings used by the MCDM models suggest that the models have varied assumptions on the importance of time-related variables in stock market prediction. Models like Equal Weighting and MEREC are impartial in their assumptions, assuming that each day is of equal importance, as may be the case if the temporal pattern is not well-defined. Conversely, techniques like CILOS, CRITIC, or SECA make it possible to combine importance over time through the use of weights on different elements of information. However, it does seem to focus on certain days in the WENSLO approach.

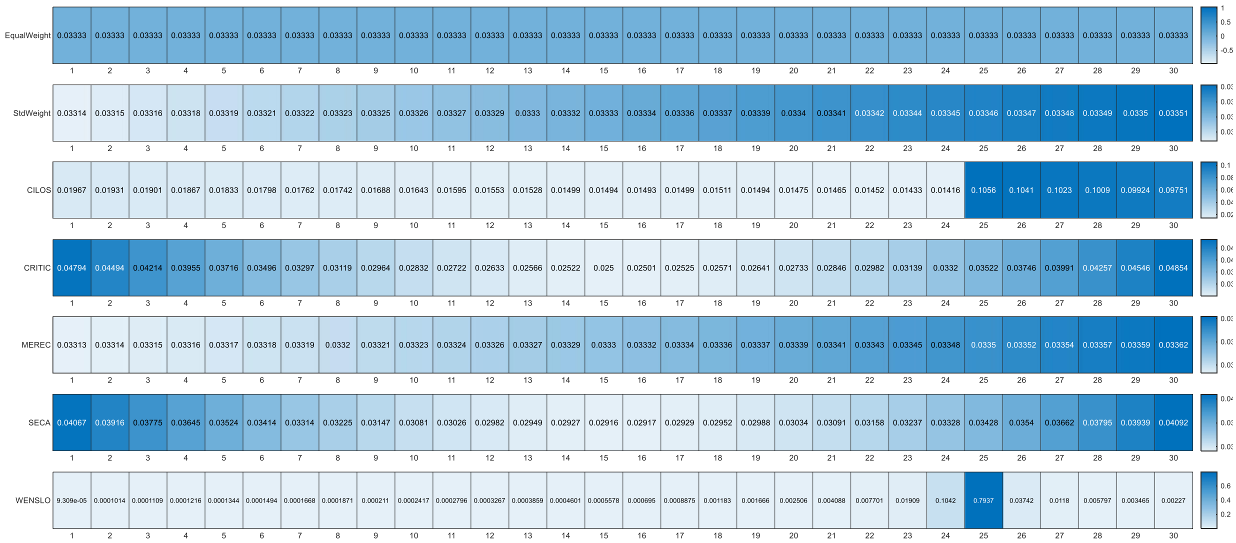
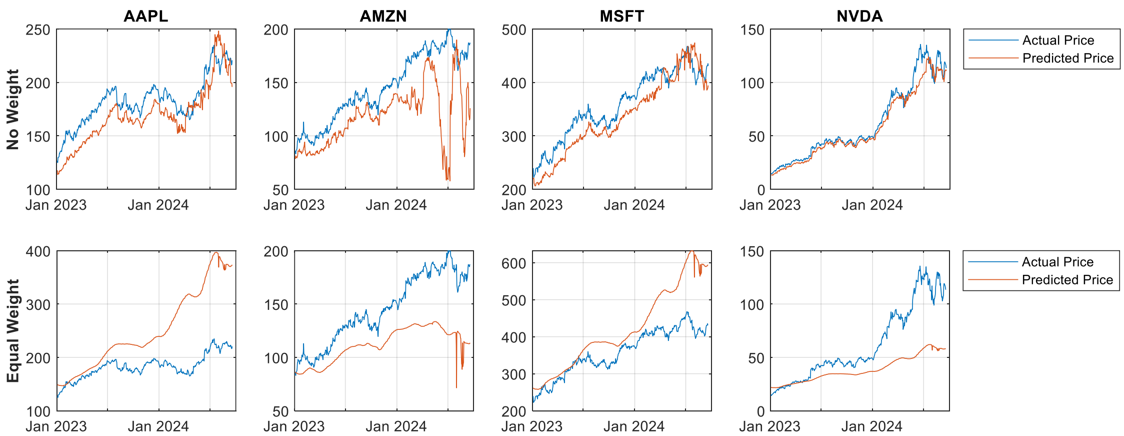


Fig. 5. Heatmap of weights from different weighting schemes

The predicted prices of the model and the actual prices for the test set, under different weight schemes for four different stocks, are presented as time series in Fig. 6.



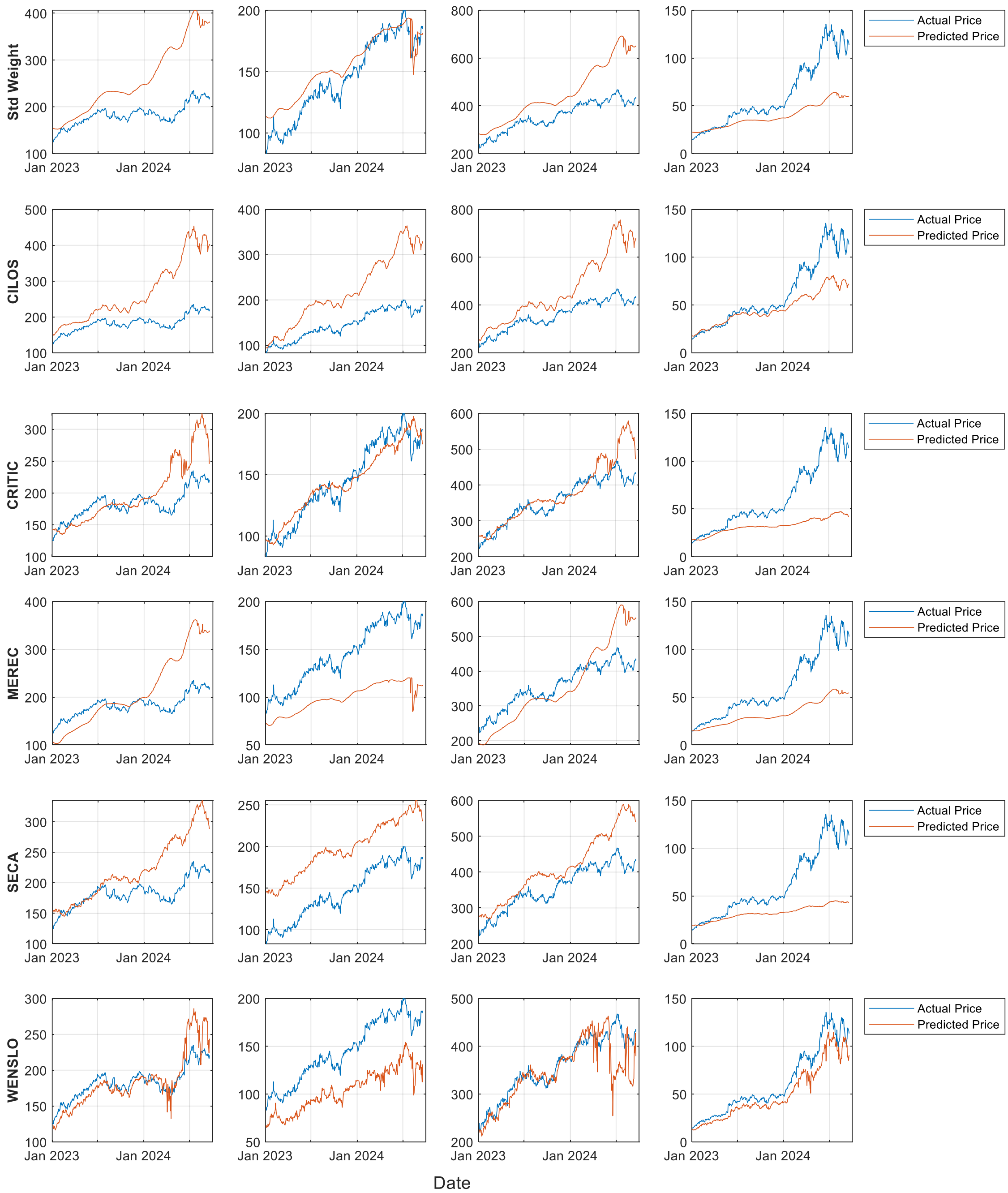


Fig. 6. Actual and predicted prices for test set

Figure 6 illustrates the actual versus predicted price values for different stocks under various weighting schemes. This comparison allows for a clear evaluation of the model's predictive ability. The graphical representation of the findings has great importance because it shows how well the model can detect hidden trends, apart from its strengths and limitations. Moreover, it enhances the transparency and interpretability of the results for a broader academic and professional audience.

Findings from comparisons of actual and predicted values under various weighting schemes (Figure 7) have several implications with respect to the model's accuracy and robustness to input weighting. The following are implications of the breakdown of the schemes:

No Weight: Actual–predicted differences are mostly positive across stocks, indicating a general overestimation tendency when no weighting is applied.

Equal Weight: Differences are negative for AAPL and MSFT but positive for AMZN and NVDA, suggesting that uniform weighting fails to capture stock-specific patterns and leads to inconsistent biases.

Standard Deviation Weight: AAPL and MSFT have a weakening negative trend, whereas AMZN and NVDA have a positive trend, indicating that volatility-weighting has a favorable effect on stocks that have a high degree of volatility but could have a distorting effect on stocks that are more stable.

CILOS: AAPL, AMZN, and MSFT display decreasing negative trends, while NVDA shows a positive increasing pattern, indicating a differentiated impact that favors growth-oriented stocks.

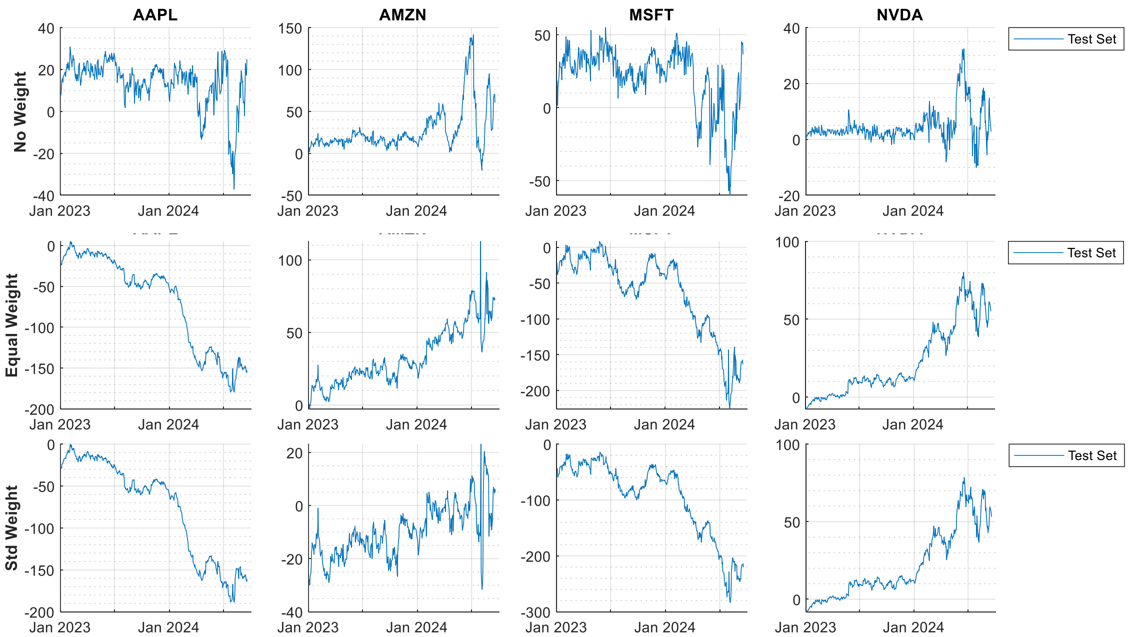
CRITIC: Only NVDA exhibits a clear positive trend, while other stocks show no consistent pattern, suggesting limited generalizability across diverse stocks.

MEREC: AAPL and MSFT transition from positive to negative differences over time, indicating sensitivity to temporal momentum changes and potential time-dependent bias.

SECA: Negative trends dominate for AAPL, AMZN, and MSFT, whereas NVDA shows a positive increasing trend, implying effectiveness mainly for stocks with strong upward momentum.

WENSLO: All stocks have positive trends with predominantly increasing patterns in terms of stable initial performance and strong growth in later periods, which means that WENSLO provides the most optimal performance for different stocks' characteristics.

The results indicate that the choice of weighting scheme significantly influences prediction accuracy and model performance across stocks. Each method has its own alignment pattern based on various characteristics of stocks, thereby emphasizing the importance of proper consideration in choosing an appropriate weight method for the stock being considered. It can also be noted that WENSLO and standard deviation methods have the capability of aligning growth and stability, while CRITIC methods might need parameterization for universal applicability. The results of this study provide significant insights for improvement of predictive models used for stock price prediction and show that individualizing weights for input variables can improve the robustness of predictive models.



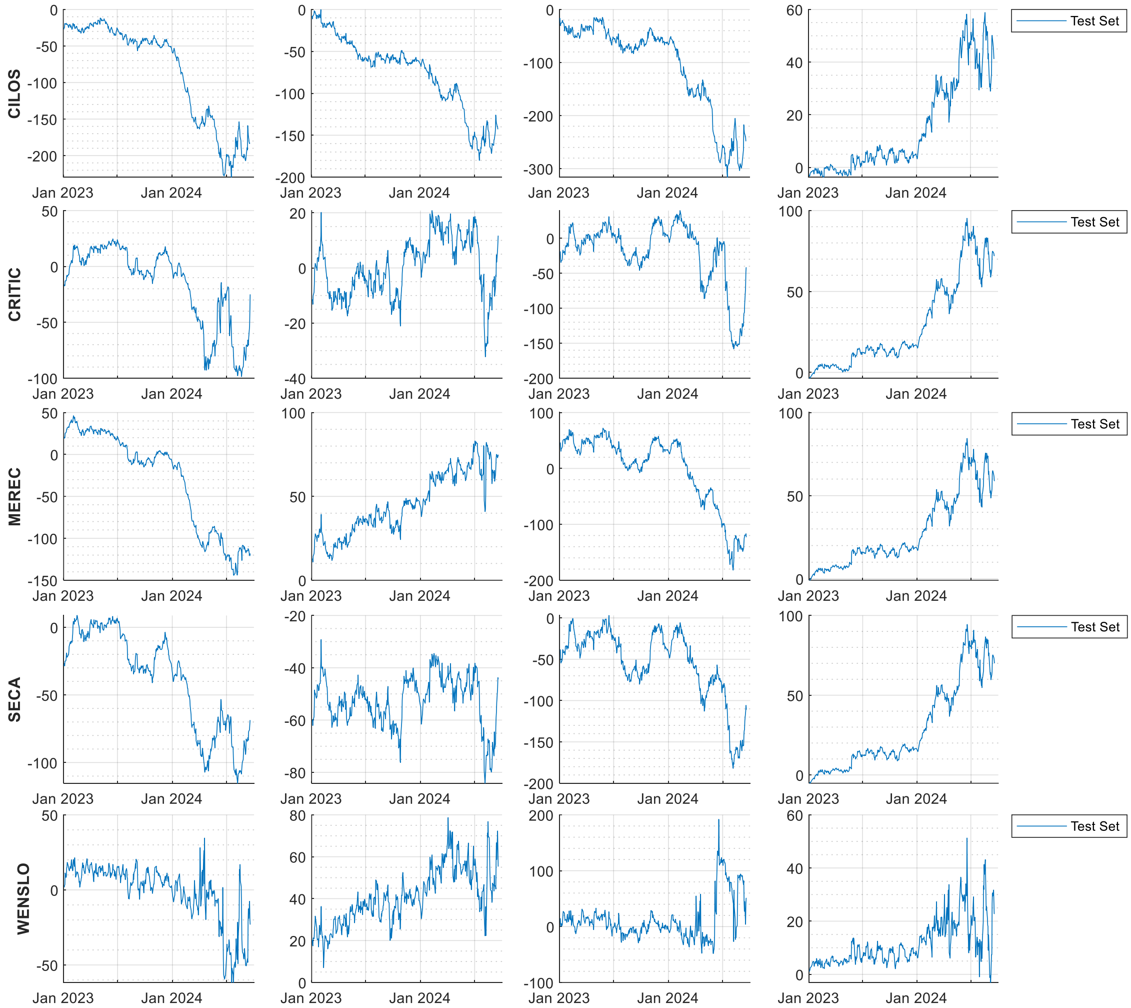


Fig. 7. Error for test set (Actual–predicted)

Table 5 summarizes test-set performance for AAPL, AMZN, MSFT, and NVDA under different weighting schemes using RMSE, MAPE, and Hit Rate. The No Weight scheme shows competitive RMSE, particularly for AAPL and NVDA, while WENSLO and CRITIC perform strongly, with CRITIC yielding the lowest RMSE for AMZN. CILOS and Standard Deviation weighting produce higher RMSE values, especially for MSFT.

WENSLO achieves the lowest MAPE for AAPL and MSFT and consistently the highest Hit Rates across all stocks, whereas CRITIC also performs well for AMZN and MSFT. In contrast, Equal Weight and No Weight schemes generally show lower Hit Rates. Overall, WENSLO—and to a lesser extent CRITIC—provides the most consistent and balanced forecasting performance, confirming the strong impact of weighting scheme selection.

Table 5
Performance metrics of stocks in test set

		APPL	AMZN	MSFT	NVDI
RMSE	No Weight	17.4210	37.5287	29.8196	7.6543
	Equal Weight	88.6370	38.6611	85.2714	32.5361
	Std Weight	95.0228	13.8073	117.8425	31.6727
	CILOS	108.1955	87.5539	136.9109	23.3358
	CRITIC	38.9705	10.6643	49.8649	40.1175
	MEREC	65.0146	49.1582	64.2459	36.2584
	SECA	52.6707	53.4081	69.6957	40.1232
	WENSLO	19.4179	43.0843	38.8291	14.1619

Table 5
Continued

		APPL	AMZN	MSFT	NVDI
MAPE	No Weight	0.0901	0.1751	0.0798	0.0819
	Equal Weight	0.3558	0.2117	0.1641	0.2996
	Std Weight	0.3957	0.0948	0.2534	0.2935
	CILOS	0.4411	0.4830	0.2775	0.1842
	CRITIC	0.1433	0.0638	0.0843	0.3774
	MEREC	0.2540	0.3009	0.1416	0.4044
	SECA	0.2078	0.3857	0.1493	0.3740
	WENSLO	0.0749	0.2790	0.0636	0.1842
Hit Rate	No Weight	0.4895	0.4988	0.4755	0.4848
	Equal Weight	0.5082	0.4429	0.5012	0.5035
	Std Weight	0.5105	0.4872	0.5082	0.5198
	CILOS	0.5385	0.4476	0.4942	0.5455
	CRITIC	0.5105	0.5105	0.5058	0.5291
	MEREC	0.5082	0.4709	0.4942	0.5152
	SECA	0.5175	0.5105	0.5012	0.5221
	WENSLO	0.5291	0.5152	0.4802	0.5338

6.6. Evaluating financial performance of weighting schemes

An investment strategy was developed to examine the financial performance of prediction models. The investment strategy can be described as follows: the entire available capital is allocated across stocks that are forecasted to yield a positive return, with the allocation being proportional to the magnitude of the predicted returns for each stock. In this approach, only those stocks with positive projected returns are considered for investment.

The capital is distributed among the selected stocks in direct proportion to their expected returns, rather than being evenly divided. This proportional allocation ensures that stocks with higher predicted returns receive a greater share of the total capital, while stocks with lower expected returns are allocated a smaller portion.

In the calculations, the risk-free interest rate has been assumed to be 0.01. The transaction cost has been considered as 0.0001.

Table 6 summarizes the backtesting results of various financial performance metrics under different weighting schemes for investment strategies. Each row delineates key performance indicators that reveal the effectiveness of these models in the context of financial returns, risk management, and trading efficiency.

Total Return: CILOS achieves the highest total return (1.1416), followed by CRITIC (1.2193), while the No Weight strategy yields a much lower return (0.0831), highlighting the benefit of structured weighting.

Sharpe Ratio: CRITIC (0.1379) slightly outperforms CILOS (0.1392) in risk-adjusted returns, whereas No Weight shows the lowest Sharpe ratio (0.0273).

Volatility: No Weight has the lowest volatility (0.0060), while CILOS (0.0131) and CRITIC (0.0139) involve higher risk.

Average Turnover: No Weight exhibits the highest average turnover (0.0686), compared to the lower and more conservative CILOS value (0.0151).

Max Turnover: Maximum turnover is highest for No Weight (1.0000), while Equal Weight (0.9220) and CILOS (0.5000) show more moderate levels.

Average Return: CRITIC records the highest average return (0.0020), emphasizing the role of weighting selection.

Max Drawdown: CILOS experiences the largest drawdown (0.1460), whereas No Weight shows a lower drawdown (0.1184).

Average Buy and Sell Costs: CILOS incurs the highest transaction costs, with average buy (0.0917) and sell (0.0893) costs, which may reduce net profitability.

In conclusion, advanced weighting schemes significantly affect investment performance. Although CILOS and CRITIC improve returns and risk-adjusted results, they increase volatility and costs, highlighting the need to balance performance and risk. Overall, appropriate weighting selection enhances investment outcomes.

Table 6
Summaries of the backtesting

	No Weight	Equal Weight	Std Weight	CILOS	CRITIC	MEREC	SECA	WENSLO
Total Return	0.0831	1.0555	1.0227	1.1416	1.2193	0.1931	1.1610	0.3855
Sharpe Ratio	0.0273	0.1363	0.1316	0.1392	0.1379	0.0418	0.1279	0.0735
Volatility	0.0060	0.0126	0.0128	0.0131	0.0139	0.0101	0.0146	0.0106
Average Turnover	0.0686	0.0417	0.0196	0.0151	0.0536	0.0270	0.0276	0.0901
Max Turnover	1.0000	0.9220	0.5000	0.5000	0.5424	0.5000	0.5000	1.0000
Average Return	0.0002	0.0018	0.0017	0.0019	0.0020	0.0005	0.0019	0.0008
Max Drawdown	0.1184	0.1266	0.1296	0.1460	0.1399	0.1615	0.1447	0.1176
Average Buy Cost	0.0705	0.0618	0.0304	0.0235	0.0917	0.0309	0.0433	0.1042
Average Sell Cost	0.0706	0.0595	0.0281	0.0212	0.0893	0.0286	0.0410	0.1019

The portfolio value of an investor who began trading with an initial portfolio value of 10,000 USD and followed the investment strategy outlined above over a period of 430 trading days is illustrated in Figure 8. It clearly demonstrates the varying outcomes of different weighting strategies applied to the initial investment over the specified trading period.

The No Weight strategy yields a modest portfolio value of USD 10,916.11, indicating limited gains without weighting. Equal Weight (USD 21,279.29) and Standard Weight (USD 20,960.73) substantially improve performance, though the latter underperforms Equal Weight. CILOS (USD 22,230.51) and SECA (USD 22,629.92) deliver strong outcomes, while CRITIC achieves the highest portfolio value (USD 23,145.06). In contrast, MEREC (USD 12,194.47) and WENSLO (USD 14,191.52) show relatively weaker performance.

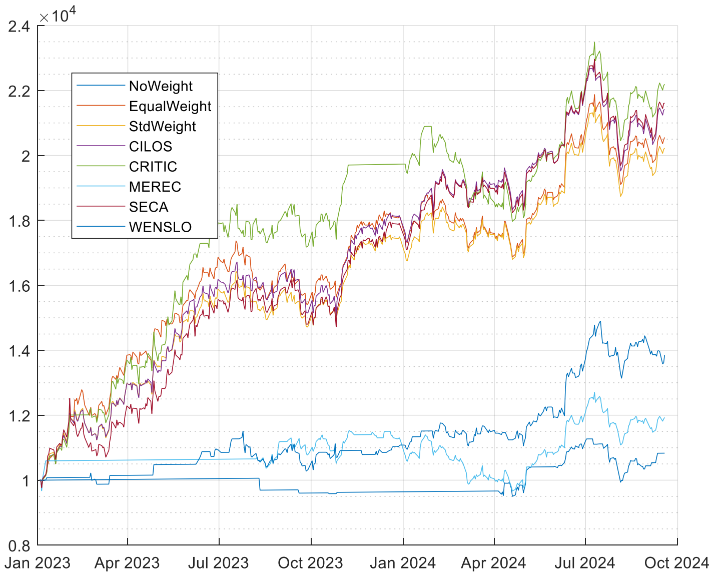


Fig. 8. Equity curves for weight schemes

In this study, the test set comprises 430 trading days. The performance of the models was assessed as the cumulative length of the test set gradually increased, reaching up to 430 days. Specifically, the performance was analyzed when the test set length was 3 days, 4 days, 5 days, and so on, up to 430 days. Figure 9 illustrates the performance of the stocks at different test set lengths when applying the WENSLO weighting method.

Considering hit rate and MAPE as performance indicators, after initial fluctuations in smaller test sets, the performance of the models stabilizes. Beyond this point, there are no significant changes until the test set reaches 430 days. However, the RMSE increases in direct proportion to the size of the test dataset. The larger the test dataset, the larger the RMSE values, hence the lower the accuracy of the predictions made by the models. This is well demonstrated by the MSFT stock, where the RMSE increases rapidly in the final 60 days of the study.

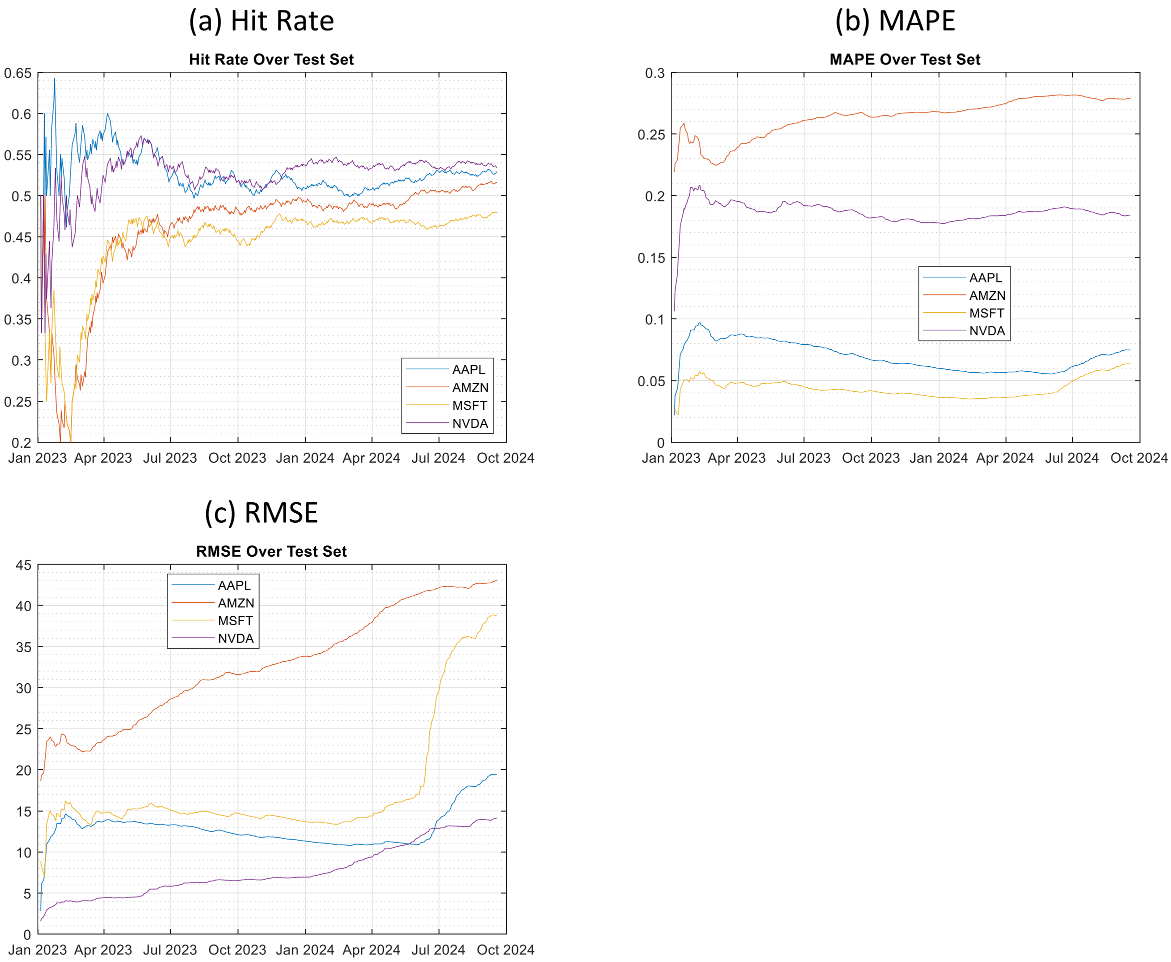


Fig. 9. Performance of the transformer neural network over the different length of test sets

6.7. Hypothesis testing for performance comparison

In this section, the presence of any statistically significant differences in performance metrics is examined using one-way ANOVA. Specifically, the analysis focuses on whether there are differences in variables such as RMSE, MAPE, Hit rate, and Total return. The general form of the hypotheses tested is as follows:

Null Hypothesis (H_0): The mean values across all categories are equal.

$$H_0: \mu_{no\ weight} = \mu_{equal} = \mu_{std} = \mu_{CILOS} = \mu_{CRITIC} = \mu_{MEREC} = \mu_{SECA} = \mu_{WENSLO}$$

Alternative Hypothesis (H_1): At least one category has a mean that is different from the others.

$$H_1: At\ least\ one\ \mu_i\ is\ different\ from\ the\ others$$
$$i = No\ weight, equal, std, CILOS, CRITIC, MEREC, SECA, WENSLO$$

In hypothesis testing, performance values from different test sets were used. Transformer neural network models, trained with input sets weighted by each weight set in the weighting scheme, were evaluated for performance over test sets of varying lengths—3 days, 4 days, 5 days, ..., up to 430 days, as outlined in the previous section. A total of 428 performance values were recorded. These 428 performance values were then used in the ANOVA analysis.

6.8. Hypothesis test results on the differences in RMSE

The ANOVA results for RMSE values calculated under different weighting schemes are presented in Table 7, the box plot of the data is shown in Figure 10. The results of the ANOVA analysis for the Root Mean Square Error (RMSE) values across different weighting schemes show a significant difference between the groups. The F-value of 380.12, with a corresponding p-value of 0, indicates that there are statistically significant differences in the RMSE values between the various weighting methods (p < 0.05).

Table 7
ANOVA results for RMSE

Source	SS	Df	MS	F	P
Groups	937535.5	7	133933.6	380.12	0
Error	4822958.8	13688	352.3		
Total	5760494.3	13695			

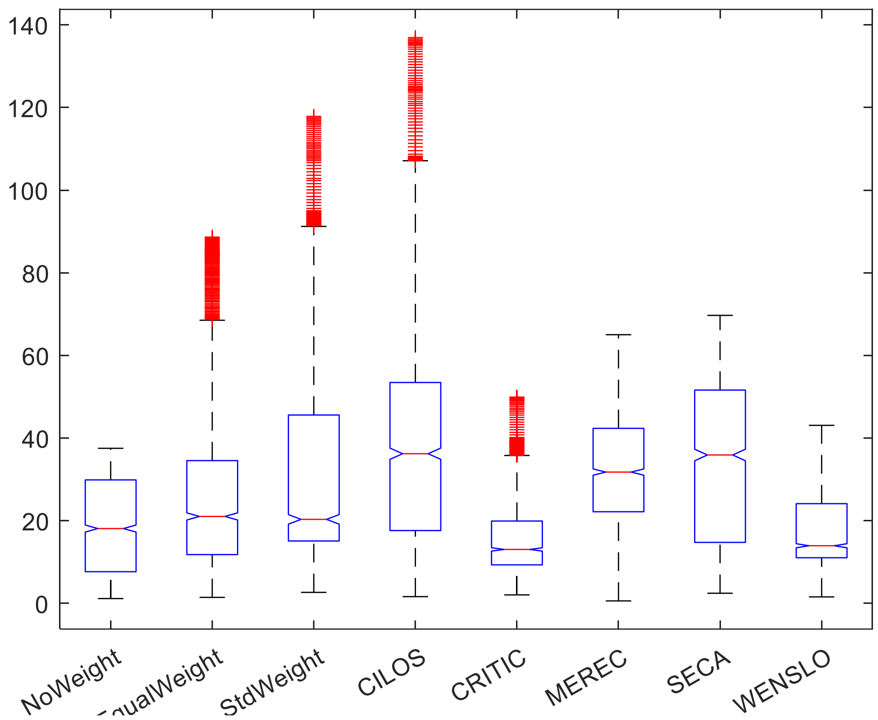


Fig. 10. ANOVA results for RMSE

The post hoc analysis (Table 8) provides a detailed comparison between the different weighting methods, offering insights into which methods yield significantly different RMSE values.

i. NoWeight vs. EqualWeight: The comparison shows a significant difference in RMSE values, with the "NoWeight" method performing worse than the "EqualWeight" method (difference = -8.15451, p < 0.01).

ii. NoWeight vs. StdWeight and CILOS: Both the "StdWeight" and "CILOS" methods outperform the "NoWeight" method significantly, with large differences in RMSE values (StdWeight: -14.3233, $p < 0.01$; CILOS: -21.69, $p < 0.01$).

iii. NoWeight vs. CRITIC and WENSLO: Interestingly, the "CRITIC" method performs worse than "NoWeight" (difference = 2.55926, $p < 0.01$), while "WENSLO" shows no significant difference compared to "NoWeight" (difference = 0.46627, $p = 0.9962$).

iv. EqualWeight vs. Other Methods: The "EqualWeight" method is outperformed by "CRITIC" and "WENSLO" ($p < 0.01$), while it outperforms "StdWeight," "CILOS," and "SECA" with statistically significant differences ($p < 0.01$).

v. CRITIC vs. Other Methods: The "CRITIC" method consistently outperforms other methods, particularly when compared to "MEREC," "SECA," and "WENSLO" ($p < 0.05$).

vi. Complexity of MEREC and SECA: Both "MEREC" and "SECA" exhibit mixed results. Neither shows significant differences between each other ($p = 0.877$), and both are outperformed by other methods, particularly "CRITIC" and "WENSLO."

Overall, the post hoc comparisons indicate that "CRITIC" and "WENSLO" are generally superior weighting methods in terms of RMSE, while methods like "MEREC" and "SECA" offer less competitive results. The analysis highlights the importance of selecting appropriate weighting schemes to optimize model performance.

Table 8
Post hoc comparison RMSE

Group	Control group	Lower Limit	Difference	Upper limit	P value
NoWeight	EqualWeight	-10.0991	-8.1545	-6.2100	0.000000
NoWeight	StdWeight	-16.2679	-14.3233	-12.3788	0.000000
NoWeight	CILOS	-23.6345	-21.6900	-19.7454	0.000000
NoWeight	CRITIC	0.6147	2.5593	4.5038	0.001705
NoWeight	MEREC	-15.6919	-13.7474	-11.8028	0.000000
NoWeight	SECA	-16.5615	-14.6170	-12.6724	0.000000
NoWeight	WENSLO	-1.4783	0.4663	2.4108	0.996225
EqualWeight	StdWeight	-8.1134	-6.1688	-4.2243	4.13E-22
EqualWeight	CILOS	-15.4800	-13.5355	-11.5909	0.000000
EqualWeight	CRITIC	8.7692	10.7138	12.6583	0.000000
EqualWeight	MEREC	-7.5374	-5.5929	-3.6483	1.56E-17
EqualWeight	SECA	-8.4070	-6.4625	-4.5179	0.000000
EqualWeight	WENSLO	6.6762	8.6208	10.5653	0.000000
StdWeight	CILOS	-9.3112	-7.3667	-5.4221	0.000000
StdWeight	CRITIC	14.9380	16.8826	18.8271	0.000000
StdWeight	MEREC	-1.3686	0.5759	2.5205	0.986354
StdWeight	SECA	-2.2382	-0.2936	1.6509	0.999814
StdWeight	WENSLO	12.8451	14.7896	16.7341	0.000000
CILOS	CRITIC	22.3047	24.2492	26.1938	0.000000
CILOS	MEREC	5.9981	7.9426	9.8871	0.000000
CILOS	SECA	5.1285	7.0730	9.0176	0.000000
CILOS	WENSLO	20.2117	22.1563	24.1008	0.000000
CRITIC	MEREC	-18.2512	-16.3066	-14.3621	0.000000
CRITIC	SECA	-19.1208	-17.1762	-15.2317	0.000000
CRITIC	WENSLO	-4.0375	-2.0930	-0.1484	0.024511
MEREC	SECA	-2.8141	-0.8696	1.0750	0.877256
MEREC	WENSLO	12.2691	14.2137	16.1582	0.000000
SECA	WENSLO	13.1387	15.0832	17.0278	0.000000

6.9. Hypothesis test results on the differences in MAPE

The ANOVA results for MAPE values calculated under different weighting schemes are presented in Table 9, the box plot of the data is shown in Figure 11. The ANOVA results reveal a significant effect of different weighting schemes on the MAPE values, with a p-value of 0.00 (F(7, 13688) = 440.7). The very low p-value suggests that there are statistically significant differences between the groups. The Sum of Squares (SS) for the groups is 26.277, while the total SS is 142.874, indicating that a notable portion of the variance in MAPE is attributable to the group differences. The within-group variance is relatively small, with an error mean square (MS) of 0.00852, underscoring the reliability of the findings

Table 9
ANOVA results for MAPE

Source	SS	Df	MS	F	P
Groups	26.277	7	3.7539	440.7	0
Error	116.596	13688	0.00852		
Total	142.874	13695			

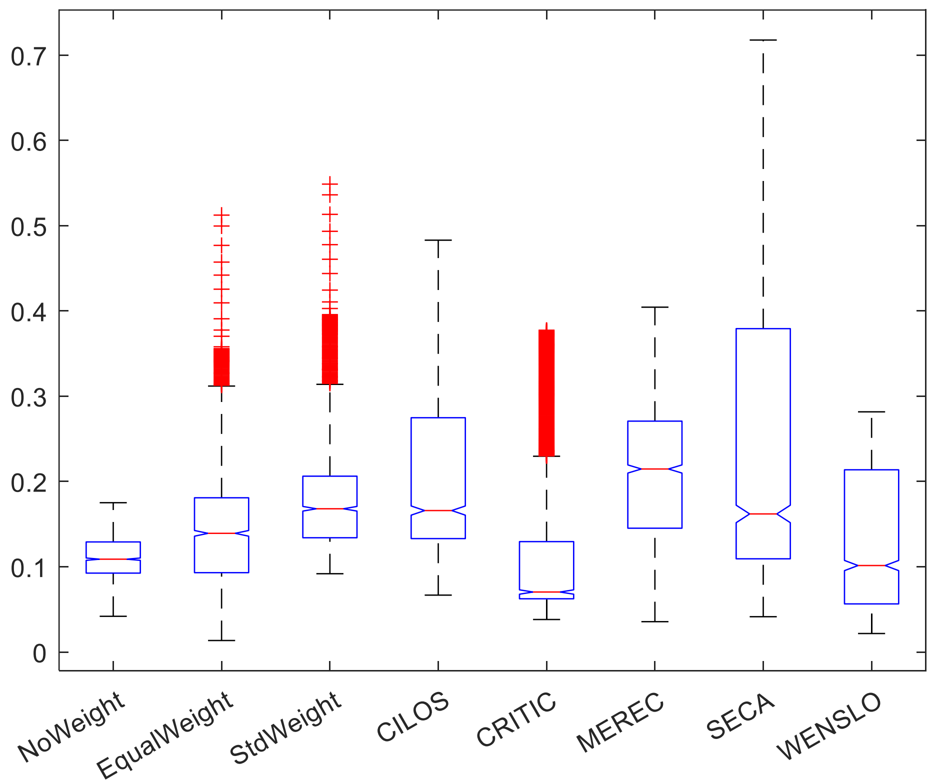


Fig. 11. Box plot for MAPE

The post-hoc analysis (Table 10) reveals statistically significant differences in MAPE between most weighting schemes, particularly when compared with the unweighted (NoWeight) approach. Notably, CILOS shows a significant improvement over NoWeight (CI: −0.10314 to −0.08402, p = 0), indicating lower MAPE values, whereas CRITIC does not differ significantly from NoWeight (p = 0.999994), suggesting comparable performance. WENSLO also significantly outperforms the unweighted scheme (p = 8.26E−22).

Furthermore, CRITIC demonstrates significant advantages over several methods, including MEREC and SECA (p = 0), confirming its robustness across multiple comparisons. Overall, while CILOS, MEREC, and WENSLO significantly improve prediction accuracy relative to NoWeight, CRITIC stands out for its consistently strong and stable performance, making it a promising weighting scheme for minimizing MAPE.

Table 10
The post-hoc analysis results

Group	Control group	Lower Limit	Difference	Upper limit	P value
NoWeight	EqualWeight	-0.04823	-0.03866	-0.02910	0.00E+00
NoWeight	StdWeight	-0.07903	-0.06947	-0.05991	0.000000
NoWeight	CILOS	-0.10314	-0.09358	-0.08402	0.000000
NoWeight	CRITIC	-0.00869	0.00088	0.01044	0.999994
NoWeight	MEREC	-0.11618	-0.10662	-0.09706	0.000000
NoWeight	SECA	-0.12771	-0.11815	-0.10859	0.000000
NoWeight	WENSLO	-0.03974	-0.03017	-0.02061	8.26E-22
EqualWeight	StdWeight	-0.04037	-0.03081	-0.02124	0.00E+00
EqualWeight	CILOS	-0.06447	-0.05491	-0.04535	0.000000
EqualWeight	CRITIC	0.02998	0.03954	0.04910	0.00E+00
EqualWeight	MEREC	-0.07751	-0.06795	-0.05839	0.00E+00
EqualWeight	SECA	-0.08905	-0.07949	-0.06992	0.000000
EqualWeight	WENSLO	-0.00107	0.00849	0.01805	1.25E-01
StdWeight	CILOS	-0.03367	-0.02411	-0.01455	3.49E-13
StdWeight	CRITIC	0.06078	0.07035	0.07991	0.000000
StdWeight	MEREC	-0.04671	-0.03715	-0.02759	0.000000
StdWeight	SECA	-0.05824	-0.04868	-0.03912	0.000000
StdWeight	WENSLO	0.02974	0.03930	0.04886	0.000000
CILOS	CRITIC	0.08489	0.09445	0.10401	0.000000
CILOS	MEREC	-0.02260	-0.01304	-0.00348	0.000933
CILOS	SECA	-0.03413	-0.02457	-0.01501	9.89E-14
CILOS	WENSLO	0.05384	0.06340	0.07296	0.000000
CRITIC	MEREC	-0.11705	-0.10749	-0.09793	0.000000
CRITIC	SECA	-0.12859	-0.11903	-0.10946	0.000000
CRITIC	WENSLO	-0.04061	-0.03105	-0.02149	0.000000
MEREC	SECA	-0.02109	-0.01153	-0.00197	0.006246
MEREC	WENSLO	0.06688	0.07644	0.08600	0.000000
SECA	WENSLO	0.07841	0.08798	0.09754	0.000000

6.10. Hypothesis test results on the differences in hit rate

The ANOVA results for Hit rate values calculated under different weighting schemes are presented in Table 11, the box plot of the data is shown in Figure 12. The ANOVA analysis and post hoc tests for the hit rate values across different weighting schemes reveal significant differences in performance. The ANOVA results indicate a highly significant effect of the weighting schemes on the hit rate, with a F-value of 188.96 and a p-value of 1.77e-268, which is far below conventional significance thresholds. This strongly suggests that the choice of weighting scheme plays a critical role in determining model

Table 11
ANOVA results for Hit Rate

Source	SS	Df	MS	F	P
Groups	3.1026	7	0.44323	188.96	1.77e-268
Error	32.1068	13688	0.00235		
Total	35.2094	13695			

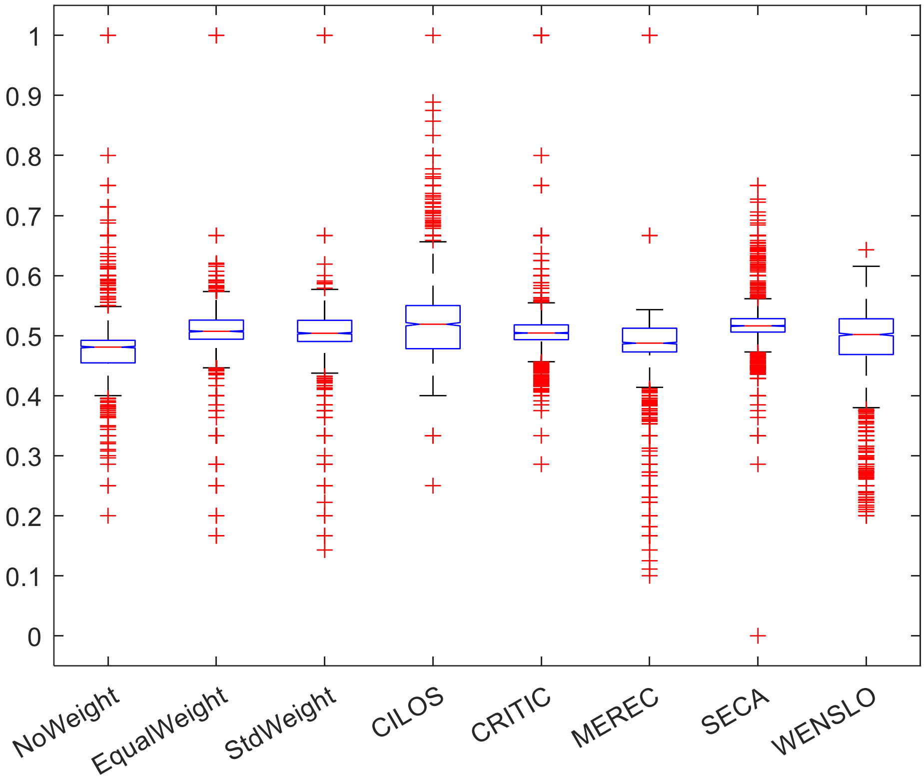


Fig. 12. Total return under different weighting schemes

The post hoc analysis (Table 12) breaks down these differences and provides pairwise comparisons between weighting schemes.

i. NoWeight vs. Other Schemes: The "NoWeight" method generally shows a lower hit rate than most other methods. For example, it performs significantly worse than the "CILOS" method, with a difference of -0.043 ($p < 0.001$), indicating that methods like CILOS lead to a notably better hit rate. The "NoWeight" method also underperforms compared to "MEREC" ($p = 0.03$), although the difference is smaller, suggesting MEREC is slightly more effective but not as significantly distinct as others.
ii. EqualWeight vs. Other Schemes: The "EqualWeight" method shows no significant difference from "StdWeight" and "CRITIC" ($p = 0.425$ and $p = 0.998$, respectively), indicating these methods produce similar hit rates. However, the "EqualWeight" method significantly underperforms when compared to "MEREC" and "WENSLO" ($p < 0.001$), both of which provide higher hit rates.
iii. CILOS vs. Other Schemes: The "CILOS" method demonstrates strong performance, significantly outperforming the "NoWeight" and "StdWeight" methods ($p < 0.001$ for both). However, when compared to "MEREC" and "WENSLO," the "CILOS" method also shows a notable difference, with "MEREC" performing even better (difference = 0.038, $p < 0.001$).

Key Takeaways:

i. Weighting schemes such as "CILOS," "MEREC," and "WENSLO" consistently outperform the "NoWeight" method and several other schemes, which indicate their superiority in achieving a higher hit rate.

ii. The "MEREC" scheme seems to offer the most substantial improvements in hit rate compared to other methods, making it one of the more promising approaches for stock price prediction.
iii. Selecting the appropriate weighting scheme can have a profound impact on model performance, as demonstrated by the significant performance differences observed between various methods.

These findings reinforce the importance of using advanced weighting techniques like MCDM methods to enhance forecasting models, especially when accurate prediction rates are crucial to success.

Table 12
Post hoc comparison.

Group	Control Group	Lower Limit	Difference	Upper limit	P value
NoWeight	EqualWeight	-0.034959595	-0.029942412	-0.024925230	0.000000000
NoWeight	StdWeight	-0.031510265	-0.026493082	-0.021475900	0.000000000
NoWeight	CILOS	-0.048392108	-0.043374925	-0.038357743	0.000000000
NoWeight	CRITIC	-0.033894791	-0.028877609	-0.023860426	0.000000000
NoWeight	MEREC	-0.010304669	-0.005287486	-0.000270304	0.030448745
NoWeight	SECA	-0.046183300	-0.041166117	-0.036148935	0.000000000
NoWeight	WENSLO	-0.016599248	-0.011582066	-0.006564883	5.87E-11
EqualWeight	StdWeight	-0.001567853	0.003449330	0.008466512	0.425456854
EqualWeight	CILOS	-0.018449696	-0.013432513	-0.008415331	5.44E-15
EqualWeight	CRITIC	-0.003952379	0.001064803	0.006081986	0.998258459
EqualWeight	MEREC	0.019637743	0.024654926	0.029672108	0.000000000
EqualWeight	SECA	-0.016240888	-0.011223705	-0.006206523	2.87E-10
EqualWeight	WENSLO	0.013343164	0.018360346	0.023377529	0.000000000
StdWeight	CILOS	-0.021899026	-0.016881843	-0.011864660	0.000000000
StdWeight	CRITIC	-0.007401709	-0.002384526	0.002632656	0.838432726
StdWeight	MEREC	0.016188413	0.021205596	0.026222779	0.000000000
StdWeight	SECA	-0.019690218	-0.014673035	-0.009655853	3.44E-18
StdWeight	WENSLO	0.009893834	0.014911016	0.019928199	7.36E-19
CILOS	CRITIC	0.009480134	0.014497317	0.019514499	1.03E-17
CILOS	MEREC	0.033070256	0.038087439	0.043104622	0.000000000
CILOS	SECA	-0.002808375	0.002208808	0.007225990	0.885900922
CILOS	WENSLO	0.026775677	0.031792859	0.036810042	0.000000000
CRITIC	MEREC	0.018572940	0.023590122	0.028607305	0.000000000
CRITIC	SECA	-0.017305691	-0.012288509	-0.007271326	2.12E-12
CRITIC	WENSLO	0.01227836	0.017295543	0.022312725	0
MEREC	SECA	-0.040895814	-0.035878631	-0.030861449	0
MEREC	WENSLO	-0.011311762	-0.00629458	-0.001277397	0.003583432
SECA	WENSLO	0.024566869	0.029584051	0.034601234	0

6.11. Hypothesis test results on the differences in total return in backtesting

Table 13 reports the ANOVA results for Hit Rate values under different weighting schemes, with the corresponding box plot shown in Figure 13. The ANOVA conducted on total returns reveals statistically significant differences among the schemes ($p = 1.5e{-}297$), with a high F-value (246.41), indicating substantial between-group variance. These results confirm that weighting scheme selection has a significant effect on total returns and is therefore critical for achieving optimal forecasting performance.

Table 13
ANOVA results for total return in backtesting

Source	SS	Df	MS	F	P
Groups	72.718	7	10.3883	246.41	1.5e-297
Error	144.016	3416	0.0422		
Total	216.734	3423			

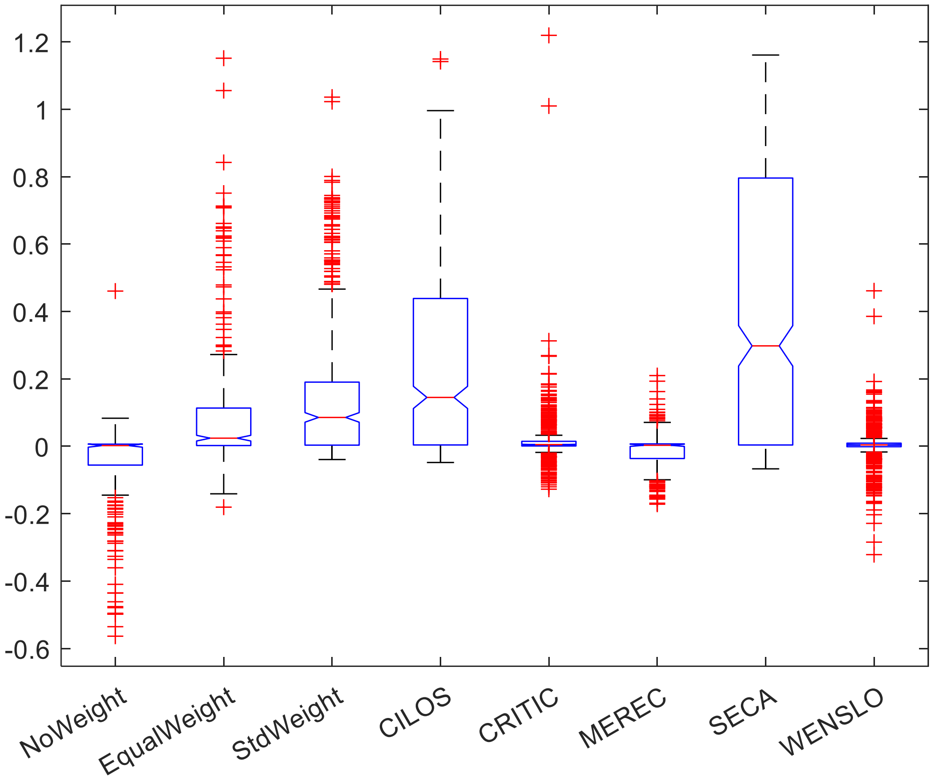


Fig. 13. Total return under different weighting schemes

The post hoc results (Table 14) show that the unweighted (NoWeight) scheme yields significantly lower total returns than most weighting methods. For example, NoWeight underperforms EqualWeight (−0.13392, p = 1.24E−21) and shows even larger differences relative to CILOS and SECA (p = 0). In contrast, CRITIC and MEREC deliver significantly higher returns. Overall, the results confirm that weighting scheme selection is a key determinant of total return in forecasting models.

Table 14
Post hoc comparison

Group	Control Group	Lower Limit	Difference	Upper limit	P value
NoWeight	EqualWeight	-0.17646	-0.13392	-0.09138	1.24E-21
NoWeight	StdWeight	-0.25156	-0.20902	-0.16648	0.000000
NoWeight	CILOS	-0.34837	-0.30583	-0.26329	0.000000
NoWeight	CRITIC	-0.10534	-0.06280	-0.02026	0.000206
NoWeight	MEREC	-0.07319	-0.03065	0.01189	0.361674
NoWeight	SECA	-0.48679	-0.44424	-0.40170	0.000000
NoWeight	WENSLO	-0.08733	-0.04479	-0.00224	0.030774
EqualWeight	StdWeight	-0.11764	-0.07510	-0.03256	2.42E-06
EqualWeight	CILOS	-0.21445	-0.17191	-0.12937	0.000000
EqualWeight	CRITIC	0.02858	0.07112	0.11366	1.11E-05
EqualWeight	MEREC	0.06073	0.10327	0.14581	3.60E-12

Table 14
Continued

Group	Control Group	Lower Limit	Difference	Upper limit	P value
EqualWeight	SECA	-0.35286	-0.31032	-0.26778	0.000000
EqualWeight	WENSLO	0.04659	0.08914	0.13168	5.58E-09
StdWeight	CILOS	-0.13936	-0.09681	-0.05427	1.22E-10
StdWeight	CRITIC	0.10368	0.14622	0.18876	0.000000
StdWeight	MEREC	0.13583	0.17837	0.22091	0.000000
StdWeight	SECA	-0.27777	-0.23523	-0.19268	0.000000
StdWeight	WENSLO	0.12169	0.16423	0.20677	0.000000
CILOS	CRITIC	0.20049	0.24303	0.28557	0.000000
CILOS	MEREC	0.23264	0.27519	0.31773	0.000000
CILOS	SECA	-0.18095	-0.13841	-0.09587	0.000000
CILOS	WENSLO	0.21851	0.26105	0.30359	0.000000
CRITIC	MEREC	-0.01039	0.03216	0.07470	0.298467
CRITIC	SECA	-0.42398	-0.38144	-0.33890	0.000000
CRITIC	WENSLO	-0.02452	0.01802	0.06056	0.905195
MEREC	SECA	-0.45614	-0.41360	-0.37106	0.000000
MEREC	WENSLO	-0.05668	-0.01414	0.02840	0.973519
SECA	WENSLO	0.35692	0.39946	0.44200	0.000000

The table provides an insightful comparison of the predictive performance of four forecasting models—Linear, LSTM, SVM, and GP—across four stocks (Apple, Amazon, Microsoft, and Nvidia) using three key performance metrics: RMSE, MAPE, and hit rate. These metrics are calculated under eight different normalization techniques: No Weight, Equal Weight, Std Weight, CILOS, CRITIC, MEREC, SECA, and WENSLO.

For RMSE, which measures the error in prediction, the Linear model generally demonstrates higher error rates compared to other models, particularly for Nvidia, where substantial deviations are observed across all normalization techniques. The SVM and LSTM models exhibit lower RMSE values, indicating better accuracy, especially with Std Weight and CILOS techniques. The GP model shows variability, with significant improvements in some cases, such as Apple and Amazon, when using specific techniques like CRITIC and No Weight.

Linear and GP models show higher MAPE, especially for NVIDIA, while SVM and LSTM are more stable; GP performs worst under WENSLO. SECA and WENSLO generally improve hit rates, particularly for SVM and LSTM, whereas Linear is inconsistent. Overall, SVM and LSTM with Std Weight or CILOS offer the most balanced performance, highlighting the importance of normalization choice (Table 15).

Table 15
Comparison of the results with different models

		LSTM				GP			
		APPL	AMZN	MSFT	NVDI	APPL	AMZN	MSFT	NVDI
RMSE	No Weight	2.5845	2.7761	5.0050	2.5447	4.3750	2.8211	12.1075	42.7519
	Equal Weight	2.8092	2.9028	5.2569	2.6490	4.3676	3.6902	8.6003	3.6839
	Std Weight	4.9223	4.6164	8.8891	4.4812	2.9723	3.0216	5.8101	2.7641
	CILOS	6.7207	5.5523	12.6502	5.0737	2.9723	3.0216	5.8101	2.7641
	CRITIC	2.6102	2.7768	5.0263	2.5310	2.9723	3.0216	5.8101	2.7641
	MEREC	4.9285	4.6202	8.9025	4.4845	45.2767	5.6777	119.7728	58.0634
	SECA	2.9222	3.0026	5.4147	2.7205	2.9723	3.0216	5.8101	2.7641
	WENSLO	6.5430	6.3053	11.7058	5.9414	29.5148	4.5793	84.5816	55.3943

Table 15
Continued

		LSTM				GP			
		APPL	AMZN	MSFT	NVDI	APPL	AMZN	MSFT	NVDI
MAPE	No Weight	0.0105	0.0148	0.0113	0.0241	0.0178	0.0148	0.0253	0.3068
	Equal Weight	0.0116	0.0153	0.0119	0.0258	0.0192	0.0196	0.0199	0.0378
	Std Weight	0.0215	0.0253	0.0211	0.0469	0.0124	0.0163	0.0130	0.0272
	CILOS	0.0306	0.0311	0.0301	0.0564	0.0124	0.0163	0.0130	0.0272
	CRITIC	0.0106	0.0146	0.0113	0.0243	0.0124	0.0163	0.0130	0.0272
	MEREC	0.0215	0.0254	0.0211	0.0470	0.1883	0.0319	0.2462	0.6359
	SECA	0.0122	0.0159	0.0122	0.0267	0.0124	0.0163	0.0130	0.0272
	WENSLO	0.0280	0.0337	0.0269	0.0615	0.1181	0.0257	0.1689	0.5674
Hit rate	No Weight	0.5082	0.4872	0.5291	0.5408	0.5175	0.4872	0.5152	0.5478
	Equal Weight	0.5245	0.4848	0.4918	0.5315	0.5175	0.5058	0.5221	0.5175
	Std Weight	0.5315	0.4685	0.5245	0.5291	0.5058	0.4895	0.5152	0.5315
	CILOS	0.4895	0.4825	0.4848	0.5385	0.5058	0.4895	0.5152	0.5315
	CRITIC	0.5105	0.4872	0.5221	0.5315	0.5058	0.4895	0.5152	0.5315
	MEREC	0.5315	0.4662	0.5221	0.5291	0.4802	0.5012	0.5035	0.4965
	SECA	0.5361	0.4755	0.4942	0.5385	0.5058	0.4895	0.5152	0.5315
	WENSLO	0.5012	0.4872	0.4732	0.5058	0.5058	0.4988	0.5058	0.5012
		Linear				SVM			
		APPL	AMZN	MSFT	NVDI	APPL	AMZN	MSFT	NVDI
RMSE	No Weight	8.2172	5.5158	16.7205	46.2153	4.1818	2.9947	11.5509	50.2569
	Equal Weight	21.6891	10.9669	35.6381	37.2760	11.1013	3.0457	48.5328	58.9428
	Std Weight	17.8096	27.9915	52.7105	41.2316	2.9424	2.9909	5.7336	2.6781
	CILOS	14.1592	10.2809	23.5552	39.8759	2.9475	3.0552	6.0478	47.6333
	CRITIC	18.9714	19.3363	35.7569	41.2526	2.8669	3.0003	5.6413	2.7530
	MEREC	31.9234	36.5531	14.3235	37.3251	3.0269	2.9902	5.9273	2.7127
	SECA	15.1906	9.1547	24.5457	40.5410	2.9457	2.9814	5.7586	2.6686
	WENSLO	48.8724	49.8169	51.5030	42.1929	3.0180	3.0529	6.9532	154.4758
MAPE	No Weight	0.0315	0.0319	0.0370	0.3999	0.0172	0.0158	0.0244	0.3539
	Equal Weight	0.1081	0.0695	0.0742	0.2917	0.0328	0.0162	0.0814	0.6006
	Std Weight	0.0826	0.1329	0.1208	0.3442	0.0122	0.0158	0.0130	0.0263
	CILOS	0.0607	0.0563	0.0590	0.3281	0.0123	0.0163	0.0134	0.2086
	CRITIC	0.0800	0.1134	0.0866	0.3573	0.0117	0.0160	0.0126	0.0267
	MEREC	0.1481	0.2005	0.0314	0.2869	0.0128	0.0160	0.0135	0.0265
	SECA	0.0738	0.0515	0.0526	0.3330	0.0123	0.0159	0.0131	0.0261
	WENSLO	0.2437	0.3015	0.1253	0.3929	0.0122	0.0161	0.0148	0.7279
Hit rate	No Weight	0.5315	0.4755	0.5058	0.5012	0.4965	0.4825	0.5175	0.5152
	Equal Weight	0.5268	0.4965	0.4942	0.5245	0.4918	0.4895	0.5385	0.5035
	Std Weight	0.5431	0.4615	0.4779	0.5175	0.5105	0.4779	0.5198	0.5128
	CILOS	0.4918	0.4779	0.4918	0.5291	0.5082	0.4755	0.5035	0.5082
	CRITIC	0.5268	0.5058	0.4825	0.5268	0.5058	0.4779	0.5175	0.5198
	MEREC	0.4942	0.5082	0.5012	0.5221	0.5012	0.4965	0.5198	0.5268
	SECA	0.5571	0.4825	0.4755	0.5291	0.4942	0.4918	0.5221	0.5198
	WENSLO	0.5035	0.4825	0.4779	0.5361	0.5128	0.4872	0.5105	0.5128

Across AAPL, AMZN, MSFT, and NVDA, the Transformer model consistently outperforms LSTM, GP, Linear, and SVM, achieving lower RMSE and MAPE and higher hit rates across weighting schemes. Its ability to capture complex temporal dependencies makes it the most accurate and reliable approach for stock price forecasting among the models tested.

7. Conclusion

Given the complexities and volatility inherent in stock markets, conventional forecasting methods frequently fail to yield accurate predictions. This study aims to propose a novel approach for stock price prediction of large-cap companies, specifically Apple, Microsoft, NVIDIA, and Amazon, which are publicly traded in the large-cap sector of the US market. This approach involves integrating objective weights obtained from MCDM methods such as SD, MEREC, SECA, CRITIC, CILOS, and WENSLO into a Transformer neural network model. Furthermore, the model was also evaluated in a scenario without any weighting. Each of these weighting techniques was applied to the dataset during the training process, with the model being trained independently for each approach.

The findings of this study underscore the critical role of objective weighting in enhancing the accuracy of stock price forecasting. By employing MCDM techniques, we have demonstrated that assigning appropriate weights to input features significantly improves the predictive performance of stock price models, particularly when utilizing advanced machine learning methodologies such as transformer neural networks. Overall, this study lays the groundwork for an innovative approach to stock price forecasting, emphasizing the value of objective weighting in improving predictive outcomes. Future research could explore additional MCDM techniques such as Entropy, AHP, and LOPCOW, or integrate other machine learning models to further enhance the robustness and applicability of the findings.

Acknowledgments

The authors would like to acknowledge the valuable comments and insightful feedback provided by all anonymous reviewers, which have helped us enhance the quality of the current study.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Billah, M., Waheed, S., & Hanifa, A. (2017). Stock market prediction using an improved training algorithm of neural network. *2016 2nd International Conference on Electrical, Computer and Telecommunication Engineering (ICECTE)*, 1-4. https://doi.org/10.1109/ICECTE.2016.7879611

[2] Upadhyay, A., & Bandyopadhyay, G. (2012). Forecasting stock performance in Indian market using multinomial logistic regression. *Journal of Business Studies Quarterly, 3*(3), 16–39.

[3] Hu, Z., Liu, W., Bian, J., Liu, X., & Liu, T.-Y. (2018). Listening to chaotic whispers: A deep learning framework for news-oriented stock trend prediction. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 261–269. https://doi.org/10.1145/3159652.3159690

[4] Gunduz, H., Yaslan, Y., & Cataltepe, Z. (2017). Intraday prediction of Borsa Istanbul using convolutional neural networks and feature correlations. *Knowledge-Based Systems, 137*, 138–148. https://doi.org/10.1016/j.knosys.2017.09.023

[5] Li, L., Leng, S., Yang, J., & Yu, M. (2016). Stock market autoregressive dynamics: A multinational comparative study with quantile regression. *Mathematical Problems in Engineering, 2016*, 1285768. https://doi.org/10.1155/2016/1285768

[6] Zhang, M., Jiang, X., Fang, Z., Zeng, Y., & Xu, K. (2019). High-order hidden Markov model for trend prediction in financial time series. *Physica A: Statistical Mechanics and Its Applications, 517*, 1–12. https://doi.org/10.1016/j.physa.2018.10.050

[7] Song, Y., & Lee, J. (2020). Importance of event binary features in stock price prediction. *Applied Sciences, 10*(5), 1597. https://doi.org/10.3390/app10051597

[8] Gao, Q. (2016). *Stock market forecasting using recurrent neural network* [Doctoral dissertation, University of Missouri].

[9] Rather, A. M., Agarwal, A., & Sastry, V. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications, 42*(6), 3234–3241. https://doi.org/10.1016/j.eswa.2014.12.003

[10] Yu, X., & Li, D. (2021). Important trading point prediction using a hybrid convolutional recurrent neural network. *Applied Sciences, 11*(9), 3984. https://doi.org/10.3390/app11093984
[11] Li, C., & Qian, G. (2022). Stock price prediction using a frequency decomposition based GRU transformer neural network. *Applied Sciences, 13*(1), 222. https://doi.org/10.3390/app13010222
[12] Gandhmal, D. P., & Kumar, K. (2019). Systematic analysis and review of stock market prediction techniques. *Computer Science Review, 34*, 100190. https://doi.org/10.1016/j.cosrev.2019.08.001
[13] Zardari, N. H., Ahmed, K., Shirazi, S. M., & Yusop, Z. B. (2015). *Weighting methods and their effects on multi-criteria decision making model outcomes in water resources management*. Springer. https://doi.org/10.1007/978-3-319-12586-2
[14] Ahn, B. S. (2011). Compatible weighting method with rank order centroid: Maximum entropy ordered weighted averaging approach. *European Journal of Operational Research, 212*(3), 552–559. https://doi.org/10.1016/j.ejor.2011.02.017
[15] Paradowski, B., Shekhovtsov, A., Bączkiewicz, A., Kizielewicz, B., & Sałabun, W. (2021). Similarity analysis of methods for objective determination of weights in multi-criteria decision support systems. *Symmetry, 13*(10), 1874. https://doi.org/10.3390/sym13101874
[16] Wang, C. N., Le, T. Q., Chang, K. H., & Dang, T. T. (2022). Measuring road transport sustainability using MCDM-based entropy objective weighting method. *Symmetry, 14*(5), 1033. https://doi.org/10.3390/sym14051033
[17] Wang, C. N., Nguyen, N. A. T., & Dang, T. T. (2023). Sustainable evaluation of major third-party logistics providers: A framework of an MCDM-based entropy objective weighting method. *Mathematics, 11*(19), 4203. https://doi.org/10.3390/math11194203
[18] Riahi, S., Bahroudi, A., Abedi, M., Lentz, D. R., & Aslani, S. (2023). Application of data-driven multi-index overlay and BWM–MOORA MCDM methods in mineral prospectivity mapping of porphyry Cu mineralization. *Journal of Applied Geophysics, 213*, 105025. https://doi.org/10.1016/j.jappgeo.2023.105025
[19] Dong, Y., Hou, J., Zhang, N., & Zhang, M. (2020). Research on how human intelligence, consciousness, and cognitive computing affect the development of artificial intelligence. *Complexity, 2020*, 1680845. https://doi.org/10.1155/2020/1680845
[20] Sarker, I. H., Furhad, M. H., & Nowrozy, R. (2021). AI-driven cybersecurity: An overview, security intelligence modeling and research directions. *SN Computer Science, 2*(3), 173. https://doi.org/10.1007/s42979-021-00557-0
[21] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539
[22] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
[23] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems, 30*. https://doi.org/10.48550/arXiv.1706.03762
[24] Thakkar, A., & Chaudhari, K. (2024). Applicability of genetic algorithms for stock market prediction: A systematic survey of the last decade. *Computer Science Review, 53*, 100652. https://doi.org/10.1016/j.cosrev.2024.100652
[25] Asadi, S., Hadavandi, E., Mehmanpazir, F., & Nakhostin, M. M. (2012). Hybridization of evolutionary Levenberg–Marquardt neural networks and data pre-processing for stock market prediction. *Knowledge-Based Systems, 35*, 245–258. https://doi.org/10.1016/j.knosys.2012.05.003
[26] Alzazah, F. S., & Cheng, X. (2020). Recent advances in stock market prediction using text mining: A survey. In F. J. García-Peñalvo (Ed.), *E-Business—Higher Education and Intelligence Applications* (pp. 1–34). Springer.
[27] Jiang, W. (2021). Applications of deep learning in stock market prediction: Recent progress. *Expert Systems with Applications, 184*, 115537. https://doi.org/10.1016/j.eswa.2021.115537
[28] Zhang, X., Zhang, Y., Wang, S., Yao, Y., Fang, B., & Yu, P. S. (2018). Improving stock market prediction via heterogeneous information fusion. *Knowledge-Based Systems, 143*, 236–247. https://doi.org/10.1016/j.knosys.2017.12.025
[29] Wang, C., Chen, Y., Zhang, S., & Zhang, Q. (2022). Stock market index prediction using deep Transformer model. *Expert Systems with Applications, 208*, 118128. https://doi.org/10.1016/j.eswa.2022.118128
[30] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval, 2*(1–2), 1–135. https://doi.org/10.1561/1500000011
[31] Yoshihara, A., Fujikawa, K., Seki, K., & Uehara, K. (2014). Predicting stock market trends by recurrent deep neural networks. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence* (pp. 759–769). Springer.
[32] Köksal, A., & Özgür, A. (2021). Twitter dataset and evaluation of transformers for Turkish sentiment analysis. *2021 29th Signal Processing and Communications Applications Conference (SIU)*, 1–4. https://doi.org/10.1109/SIU53274.2021.9477818

[33] Li, X., Wu, P., & Wang, W. (2020). Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong. *Information Processing & Management, 57*(5), 102212. https://doi.org/10.1016/j.ipm.2020.102212

[34] Qian, B., & Rasheed, K. (2007). Stock market prediction with multiple classifiers. *Applied Intelligence, 26*(1), 25–33. https://doi.org/10.1007/s10489-006-0001-7

[35] Yañez, C., Kristjanpoller, W., & Minutolo, M. C. (2024). Stock market index prediction using transformer neural network models and frequency decomposition. *Neural Computing and Applications, 36*, 7577–7597. https://doi.org/10.1007/s00521-024-09931-4

[36] Zhang, H. (2024). Stock price forecast based on improved Transformer. In *2024 3rd International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID 2024)* (pp. 170–182). Atlantis Press. https://doi.org/10.2991/978-94-6463-524-6_21

[37] Kotu, V., & Deshpande, B. (2019). Time series forecasting. In *Data Science* (2nd ed., pp. 395–445). Elsevier. https://doi.org/10.1016/B978-0-12-814761-0.00012-5

[38] Efendi, R., Arbaiy, N., & Deris, M. M. (2018). A new procedure in stock market forecasting based on fuzzy random auto-regression time series model. *Information Sciences, 441*, 113–132. https://doi.org/10.1016/j.ins.2018.02.016

[39] Rubi, M. A., Chowdhury, S., Rahman, A. A. A., Meero, A., Zayed, N. M., & Islam, K. A. (2022). Fitting multi-layer feed forward neural network and autoregressive integrated moving average for Dhaka Stock Exchange price predicting. *Emerging Science Journal, 6*(5), 1046–1061. https://doi.org/10.28991/ESJ-2022-06-05-09

[40] Deepa, N., Ganesan, K., Srinivasan, K., & Chang, C. Y. (2019). Realizing sustainable development via modified integrated weighting MCDM model for ranking agrarian dataset. *Sustainability, 11*(21), 6060. https://doi.org/10.3390/su11216060

[41] Keshavarz-Ghorabaee, M., Amiri, M., Zavadskas, E. K., Turskis, Z., & Antucheviciene, J. (2021). Determination of objective weights using a new method based on the removal effects of criteria (MEREC). *Symmetry, 13*(4), 525. https://doi.org/10.3390/sym13040525

[42] Keshavarz-Ghorabaee, M., Amiri, M., Zavadskas, E. K., Turskis, Z., & Antucheviciene, J. (2018). Simultaneous evaluation of criteria and alternatives (SECA) for multi-criteria decision-making. *Informatica, 29*(2), 265–280. https://doi.org/10.3233/INF-2018-1182

[43] Jahan, A., Mustapha, F., Sapuan, S. M., Ismail, M. Y., & Bahraminasab, M. (2012). A framework for weighting of criteria in ranking stage of material selection process. *The International Journal of Advanced Manufacturing Technology, 58*(1-4), 411–420. https://doi.org/10.1007/s00170-011-3366-7

[44] Zavadskas, E. K., & Podvezko, V. (2016). Integrated determination of objective criteria weights in MCDM. *International Journal of Information Technology & Decision Making, 15*(02), 267–283. https://doi.org/10.1142/S0219622016500036

[45] Pamucar, D., Ecer, F., Gligorić, Z., Gligorić, M., & Deveci, M. (2023). A novel WENSLO and ALWAS multicriteria methodology and its application to green growth performance evaluation. *IEEE Transactions on Engineering Management, 71*, 11992-12016. https://doi.org/10.1109/TEM.2023.3321697

[46] Chen, J., Chen, T., Shen, M., Shi, Y., Wang, D., & Zhang, X. (2022). Gated three-tower transformer for text-driven stock market prediction. *Multimedia Tools and Applications, 81*(21), 30093–30119. https://doi.org/10.1007/s11042-022-11908-1

[47] Mian, T. S. (2023). Evaluation of stock closing prices using Transformer learning. *Engineering, Technology & Applied Science Research, 13*(5), 11635–11642. https://doi.org/10.48084/etasr.6017

[48] Zhang, Q., Qin, C., Zhang, Y., Bao, F., Zhang, C., & Liu, P. (2022). Transformer-based attention network for stock movement prediction. *Expert Systems with Applications, 202*, 117239. https://doi.org/10.1016/j.eswa.2022.117239

[49] Hochreiter, S. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[50] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems, 27*, 3104–3112.