IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 29 April 2024, accepted 23 May 2024, date of publication 11 June 2024, date of current version 24 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3412975

RESEARCH ARTICLE

Predicting Medium-Term Stock Index Direction Using Constituent Stocks and Machine Learning

A. BAREKET AND B. PÂRV
Faculty of Mathematics and Computer Science, Babeş-Bolyai University, 400347 Cluj-Napoca, Romania
Corresponding author: A. Bareket (arnonb@afeka.ac.il)

ABSTRACT Predicting stock index movements is challenging due to market randomness. This paper addresses the problem of predicting medium-term stock index direction, an area with limited coverage in existing literature. We propose a novel model based on machine learning algorithms that employs relative indicators of constituent stocks within an index. The objective is to identify market entry points where the likelihood of achieving a significant return threshold is higher. To achieve this, supervised binary classifiers and other machine learning techniques have been applied, setting the class label '1' for significant index rises occurring within a defined medium term of 70 trading days and the class label '0' otherwise. Three indices were investigated: Nasdaq 100, where a significant rise was defined as a 10% increase, Dow Jones Industrial Average with an 8% increase, and the German Dax at 6%. Our investigation into different methods of utilizing constituent stocks revealed that focusing on the most weighted stocks yields the most promising results across various stock indices. The proposed model dynamically selects the most effective classifiers, SVM, KNN, Voting Classifier, and RF, tailored to varying market conditions. Employing a rolling forecast method, it utilizes the relative indicators on heavily weighted stocks and the index, demonstrating accuracy up to 0.97 and F1-scores for the '1' label up to 0.90. This enhances the ability to determine the optimal timing for market entry and, crucially, when the chances for high returns are limited. Additionally, we illuminate the conditions under which the model is most effective.

INDEX TERMS Constituent stocks within an index, machine learning algorithms, predicting medium-term stock index direction, relative indicators.

I. INTRODUCTION

Predicting stock index movements is a complex task due to the inherent randomness and the influence of numerous external factors, including financial reports, interest rate fluctuations, media opinions, and unexpected financial data. These factors can collectively shift the direction of the market. Numerous studies exploring the feasibility of forecasting market movements have demonstrated that accurate predictions may be challenging, if not impossible, to achieve or may yield only negligible benefits for specific time periods and indices [1], [2], [3], [4], [5]. This notion aligns with the Efficient Market Hypothesis (EMH) [6] and the Random Walk Theory [7], which posit that markets exhibit random walk behavior, rendering any algorithmic attempts to outperform the market futile. Nevertheless, certain studies have reported varying degrees of predictability for specific time periods, conditions, indices, or stocks [8], [9], [10], [11], [12], suggesting that market predictability may be limited to certain contexts.

Regarding prediction time horizons, the majority of existing literature focuses on forecasting the next day's value or classifying the direction of the subsequent day's movement as either upward or downward. This study, however, aims to classify medium-term periods, as this particular prediction horizon has received minimal attention in the literature. For the purposes of this discussion, we define ‘‘medium-term’’ as a period extending approximately 70 trading days into the

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

future, which equates to roughly 3.5 months. Our research seeks to identify instances when achieving a significant return above a specified threshold during this time is more probable.

In contrast to most studies that rely solely on a primary index for forecasting, our proposed model incorporates both the constituent stocks of the index and the index itself. This approach is based on the premise that aggregating the classifications of various stocks, which are highly correlated with the main index, can yield a more balanced and accurate overall prediction. Another distinguishing feature of our model is its use of relative indicators instead of absolute ones. While most studies employ indicators in absolute terms, we opted for relative indicators to identify recurring patterns independent of the index's absolute values.

Our binary classification model assigns a label of 1 to situations in which the main index achieves a substantial return above a predetermined threshold—based on the index's average standard deviation—within the subsequent 70 trading days. We are not concerned with whether the index will surpass the threshold at the end of the period; rather, our interest lies in whether this threshold is reached at any point during the period. To classify the data, our model compares the performance of various classifiers, including the K-Nearest Neighbors Classifier (KNN), Support Vector Machine Classifier (SVM), Random Forest Classifier (RF), and Voting Classifier.

The preliminary findings of this study indicate the potential for identifying favorable market entry points using our model, which uniquely incorporates constituent stocks and relative indicators on the stocks and indices. While the market's unpredictability often makes entry timing challenging, our model provides valuable insights by indicating not only when to enter but also when to avoid the market. This dual capability enhances the strategic decision-making process for investors seeking medium-term opportunities.

II. LITERATURE REVIEW

Upon review of various papers, it has been observed that most of the existing research focuses on short-term predictions. This is due to the inherent difficulty in predicting outcomes over longer time frames. As a result, many researchers have focused on short-term predictions, often limited to a few days ahead. However, this approach may not satisfy the needs of most investors who seek to stay in the markets for several months or even years. This study aims to address this gap. In this literature review, we will explore some studies that do refer to longer time frames in some way.

One study by Chen and Hao [9] classified the direction of Chinese stock markets from 2008 to 2014 across different prediction horizons ranging from one day to 30 days ahead. Common technical indicators were used as features in a weighted manner with SVM and KNN machines.

Another study by Milosevic [13] predicted the rise of 1298 different stocks in a year by using the stock prices and various financial indicators as input features. They achieved an *F1-score* of around 0.75 for selecting the right stocks.

Mittal and Nagpal [14] attempted to identify stocks in the Indian market (BSE SENSEX) that would yield good returns in the medium to long term, up to one year ahead. The study used financial indicators derived from the intrinsic health of the stock, along with price and sector indicators and other economic data, to create a regression model with fuzzy sets that provided credible gains.

Most studies in the field provide only short-term forecasts, typically the direction of the next day or several days ahead.

Zheng and Jin [4] note that studies achieving long-term prediction accuracy of around 70% often rely on company-specific information not widely available to the public. For short-term classification, such as predicting market direction for a few days, it is reported that most models typically exhibit an accuracy range of 50% to 60%. Higher reported accuracies are often attributed to problematic evaluation metrics.

Zheng and Jin [4] focused on the short term, from one day to 20 days ahead. They used 82 stocks traded on the New York Stock Exchange along with their prices, trade volume, and 17 commonly accepted technical indicators. Those were used as input features for various classifiers such as Logistic Regression, Bayesian Network, Simple Neural Network, and SVM. Their results showed that predictability decreases as the forecast horizon increases. SVM was found to be the best classifier for predicting the direction in the short term, with accuracy, which is not very high and is up to 70%. These results emphasize the complexity and difficulty of predicting market direction.

In another study, Qiu and Song [15] attempted to predict the direction of the Nikkei 225 index for the next day, between 2007 and 2013, using common indicators on the prices together with Artificial Neural Networks (ANN), which was tuned according to genetic algorithm. They reported a hit ratio (i.e., accuracy) of 81%.

Persio and Honchar [3] used different Neural Network architectures, such as Long Short-Term Memory (LSTM), ANN, and Convolutional Neural Networks (CNN), to predict the direction of the S&P500 for the next day, based on the last 30 days. Their results showed an accuracy of around 0.55 and supported the Random Walk Theory [7], emphasizing the difficulty in providing a reliable forecast.

Zhong and Enke [2] attempted to forecast the next day's movement direction of the S&P500 using around 60 different financial and economic data. Despite using a large number of explanatory variables, the accuracy was around 0.55, emphasizing the random nature of the markets.

In a study by Lei [16], researchers attempted to predict the next-day trend for five well-known indexes. They used indicators and statistics on the prices, which were used as input features. The features were then entered into a neural network along with wavelet transform on the index prices to predict the trend. The study, which used data from 2009 to 2014, found that the average precision of the predicted directions was, at best, around 66%.

In another study, Hu et al. [12] tried to predict the next day's opening direction of the Dow Jones and the S&P500 using an optimized neural network and data from Google-Trends. The sample data was from the period of 2010 to 2017, and the researchers reported a hit ratio in predicting the next day's opening direction as close to 90%.

It is worth mentioning that there are considerable differences between the above studies and the study presented here. Our study, as detailed below, focuses on the mid-term time frame and demands a gain of the index over a certain significant threshold of percentages. Additionally, our study sets the classification results not just on a target day at the end of the time frame but also allows for the target day to be any day during the medium-term of the 70 trading days ahead. Furthermore, our study employs different input features and models, and due to the fact that the market is indeed random most of the time, our model attempts to determine when it is the right time to use it and when not to use it at all.

III. COMPUTATIONAL TOOLS AND DATA RETRIEVAL

For this study, we employed Python 3.9.7, complemented by the Spyder 5.1.5 IDE environment. The classification processes utilized a range of algorithms from the sklearn package. We acquired our dataset from Yahoo Finance [17], utilizing the yfinance module in Python [18] for data retrieval. The analysis of classification outcomes was facilitated by the tools available in the sklearn.metrics package [19].

IV. DATA DESCRIPTION

A. GROUPS OF DATA

Three different groups of data were downloaded, each group consisting of the daily closing prices of one main index, followed by the daily closing prices of a list of stocks that make up that index, selected according to their relative weight in the index and their last 100 days' correlation with the index. The list contains the most weighted stocks as well as the most correlated ones. The correlation was calculated for the time point at which the test period begins, as described below, as well as the relative weights.

The first group's main index is the NASDAQ 100 (NDX). The second group's main index is Dow Jones Industrial Average (DJI), and the third group's main index is DAX (GDAXI). Table 1 displays the list of selected stocks for the NASDAQ 100, sorted by weight, corresponding to the train-test division which is detailed in the following subsection. Table 2 shows the list of selected stocks for NASDAQ 100, sorted by correlation for the last 100 days, with secondary sorting by the weight. Table 3 and Table 4 show the same for the DOW, while Table 5 and Table 6 for the DAX.

B. DATA COLLECTION PERIOD AND METHODOLOGY FOR TRAIN-TEST SPLIT

Our study sourced daily closing values from January 1, 2010, through September 15, 2022, for the NASDAQ 100, Dow Jones, and DAX indices, encompassing significant market fluctuations over approximately 12 years. The NASDAQ 100 and Dow Jones each accounted for 3198 trading days, while DAX had 3222 days in this period.

For the training and testing division, the initial bulk of the data was used for model training. The final 350 days of this period, representing about 11% of the entire dataset, were set aside for the testing phase across all three indices. This approach ensured a comprehensive analysis while adhering to computational and methodological considerations. It's important to note that certain segments of the training data were not entirely utilized for keeping analytical and computational requirements.

TABLE 1. Chosen stocks for NASDAQ 100, sorted by weight.

Stock	Symbol	Weight	Corr.
Apple Inc	AAPL	13.0	0.90
Microsoft Corp	MSFT	10.5	0.92
Amazon.com Inc	AMZN	5.2	0.83
NVIDIA Corp	NVDA	3.4	0.87
Alphabet Inc	GOOG	3.2	0.89
Alphabet Inc	GOOGL	3.1	0.89
Tesla Inc	TSLA	2.9	0.64
PepsiCo Inc	PEP	2.4	0.61
Meta Platforms Inc	META	2.2	0.66
Broadcom Inc	AVGO	2.1	0.87
Texas Instruments	TXN	1.4	0.89
QUALCOMM Inc.	QCOM	1.1	0.88
Analog Devices	ADI	0.8	0.89
Microchip Tech Inc	MCHP	0.4	0.90

TABLE 2. Chosen stocks for NASDAQ 100, sorted by correlation. (and secondary sorting by weight).

Stock	Symbol	Weight	Corr.
Microsoft Corp	MSFT	10.5	0.92
Apple Inc	AAPL	13.0	0.90
Microchip Tech Inc	MCHP	0.4	0.90
Alphabet Inc	GOOG	3.2	0.89
Alphabet Inc	GOOGL	3.1	0.89
Texas Instruments	TXN	1.4	0.89
Analog Devices	ADI	0.8	0.89
QUALCOMM Inc.	QCOM	1.1	0.88
NVIDIA Corp	NVDA	3.4	0.87
Broadcom Inc	AVGO	2.1	0.87
Amazon.com Inc	AMZN	5.2	0.83
Meta Platforms Inc	META	2.2	0.66
Tesla Inc	TSLA	2.9	0.64
PepsiCo Inc	PEP	2.4	0.61

TABLE 3. Chosen stocks for Dow Jones, sorted by weight.

Stock	Symbol	Weight	Corr.
UnitedHealth	UNH	10.3	0.64
Goldman Sachs	GS	7.4	0.84
Home Depot	HD	6.3	0.80
Amgen Inc	AMGN	5.5	0.53
McDonald's	MCD	5.2	0.63
Microsoft Corp	MSFT	4.9	0.83
Caterpillar Inc	CAT	4.5	0.75
Honeywell Int.	HON	4.2	0.88
Visa Inc	V	4.2	0.78
Travelers	TRV	3.6	0.61
American Express	AXP	2.9	0.85
Apple Inc	AAPL	2.6	0.82
JPMorgan Chase	JPM	2.6	0.81
Coca-Cola	KO	1.3	0.81
Intel Corp	INTC	0.5	0.83

TABLE 4. Chosen stocks for Dow Jones, sorted by correlation. (and secondary sorting by weight).

Stock	Symbol	Weight	Corr.
Honeywell Int.	HON	4.2	0.88
American Express	AXP	2.9	0.85
Goldman Sachs	GS	7.4	0.84
Microsoft Corp	MSFT	4.9	0.83
Intel Corp	INTC	0.5	0.83
Apple Inc	AAPL	2.6	0.82
JPMorgan Chase	JPM	2.6	0.81
Coca-Cola	KO	1.3	0.81
Home Depot	HD	6.3	0.80
Visa Inc	V	4.2	0.78
Caterpillar Inc	CAT	4.5	0.75
UnitedHealth	UNH	10.3	0.64
McDonald's	MCD	5.2	0.63
Travelers	TRV	3.6	0.61
Amgen Inc	AMGN	5.5	0.53

TABLE 5. Chosen stocks for Dax, sorted by weight.

Stock	Symbol	Weight	Corr.
Linde	LIN.DE	10.6	0.82
SAP	SAP.DE	8.0	0.87
Siemens	SIE.DE	7.0	0.91
Allianz	ALV.DE	6.6	0.88
Deutsche Tel.	DTE.DE	6.1	0.71
Airbus	AIR.DE	5.0	0.80
Bayer	BAYN.DE	4.8	0.64
Mercedes-B.	MBG.DE	4.2	0.82
BASF	BAS.DE	3.5	0.84
Munich Re	MUV2.DE	3.4	0.62
Deutsche Post AG	DPW.DE	2.9	0.86
Adidas AG	ADS.DE	2.1	0.76
Deutsche Bank	DBK.DE	1.7	0.71
Porsche	PAH3.DE	1.0	0.70

TABLE 6. Chosen stocks for Dax, sorted by correlation. (and secondary sorting by weight).

Stock	Symbol	Weight	Corr.
Siemens	SIE.DE	7.0	0.91
Allianz	ALV.DE	6.6	0.88
SAP	SAP.DE	8.0	0.87
Deutsche Post AG	DPW.DE	2.9	0.86
BASF	BAS.DE	3.5	0.84
Linde	LIN.DE	10.6	0.82
Mercedes-B.	MBG.DE	4.2	0.82
Airbus	AIR.DE	5.0	0.80
Adidas AG	ADS.DE	2.1	0.76
Deutsche Tel.	DTE.DE	6.1	0.71
Deutsche Bank	DBK.DE	1.7	0.71
Porsche	PAH3.DE	1.0	0.70
Bayer	BAYN.DE	4.8	0.64
Munich Re	MUV2.DE	3.4	0.62

C. NEW TECHNICAL INDICATORS

Technical indicators represent some type of memory since they encapsulate information from the past in their various calculations. Various studies have shown that using technical indicators can improve predictability [20], [21], [22], [23]. In this research, we created a new form of technical indicators based on relationships between existing indicators. Our rationale was that technical indicators like simple moving averages do not provide meaningful information on their own. The same values of a simple moving average can occur in different market situations, and its value alone does not contribute enough information for accurate predictions. Therefore, we developed new indicators based on relationships to make the predictability more robust for different times and situations in the market. We tested various technical indicators and their relationships and eventually developed our indicators based on the formulas below. In the formulas, n represents the time periods, and t represents the data point at period t.

The Triple Exponential Moving Average ($TEMA$) is defined below. First, we define the Exponential Moving Average (EMA) as follows:

$$EMA_t(n) = (close_t - EMA_{t-1}(n)) \cdot k + EMA_{t-1}(n) \quad (1)$$

where $k =$ the smoothing constant, equals to $\frac{2}{n+1}$, and $close_t$ is defined as the close price at period t, as Equation (7) states, and the First EMA is calculated as a simple average over the first n periods.

By denoting

$$\begin{aligned} EMA1_t(n) &= EMA_t(n) \\ EMA2_t(n) &= EMA_t(EMA1_t(n)) \\ EMA3_t(n) &= EMA_t(EMA2_t(n)) \end{aligned} \quad (2)$$

the Triple Exponential Moving Average ($TEMA$) is computed as follows:

$$TEMA_t(n) = 3 \cdot EMA1_t(n) - 3 \cdot EMA2_t(n) + EMA3_t(n) \quad (3)$$

Following this, we will define our new relative indicators:

$$MomTema_t(n, ofs) = \frac{TEMA_t(n)}{TEMA_{t-ofs}(n)} \quad (4)$$

where ofs is the offset, representing the number of periods back from the current time t.

$$RCTema_t(n) = \frac{close_t}{TEMA_t(n)} \quad (5)$$

Another indicator to be used, is defined as:

$$LogReturn_t(n) = ln(close_t) - ln(close_{t-n}) \quad (6)$$

We use the natural logarithm (ln) of the closing price to enhance classifiers' performance while mitigating the impact of outliers and normalizing the data.

In our experiments, we used only 4 different indicators, with the following parameters: $MomTema_t(300, 15)$, $MomTema_t(350, 15)$, $RCTema_t(350)$ and $LogReturn_t(50)$, as found to be most effective for our tested data-sets.

D. INDICATOR SELECTION PROCESS

Our selection of the final four indicators was driven by a strategic focus on trend responsiveness and comprehensive market dynamics. Given our goal to detect significant market rises within the next 70 trading days, we needed indicators that help the classifier understand the general trend and its potential continuation or reversal, along with the current market state relative to this trend.

We prioritized indicators that could quickly react to trend changes to enhance the probability of correctly following trends, each providing unique and complementary information perspectives to the classifier. We chose TEMA for its ability to follow trends with minimal lag, as evidenced by its mathematical formulation (see Equation 3), which reduces the lag compared to other moving averages. To capture general trend momentum in a relative way that is meaningful for the classifier, we selected MomTema indicators. RCTema was included to provide insights into price positioning relative to the trend, aiding the classifier in identifying optimal market entry points. LogReturn was used to assess recent price changes, helping predict potential trend continuations or reversals.

We conducted extensive empirical testing to validate these choices, comparing various configurations of the initially selected indicators, as well as other new relative indicators we created and other common indicators used in the field. This involved systematic testing across different indices and timeframes to ensure robustness. Each indicator and different combinations of them were evaluated across various timeframes and indices, focusing on periods where the model showed predictive power. The final combination was selected based on the best average performance during these periods. We fine-tuned parameters through optimization to enhance predictive power. Performance metrics were evaluated based on accuracy, the F1-score of the positive label (i.e., the '1' label), and Cohen's kappa. Combining the indicators showed synergy, resulting in enhanced predictive performance.

In our study, the final four indicators, $MomTema_t(300, 15)$, $MomTema_t(350, 15)$, $RCTema_t(350)$, and $LogReturn_t(50)$, demonstrated superior performance, providing better predictive accuracy and robustness across different market conditions. These choices were empirically validated, showing that the past 350 days were sufficient and effective for making accurate predictions for the next 70 trading days, which represent an addition of 20% of that period. Applying these indicators to both the selected stocks and the index itself resulted in a sufficient number of input features (12 to 16, as will be discussed later), proving effective for our model.

These indicators capture different market aspects, offering a holistic view. Slightly different periods for MomTema provide nuanced trend understanding. They perform well across stable and volatile conditions, which is crucial for medium-term forecasting. Extensive historical data testing validated their effectiveness, and they generalize well across datasets, reducing overfitting risk.

In summary, our strategic approach and empirical validation led to selecting indicators aligned with the medium-term focus, providing a robust method for predicting significant upward movements in stock indexes.

V. METHODOLOGY

A. CLASSIFICATION DEFINITION

We developed a binary classification model to predict whether the price of an index will reach a 'meaningful rise target' *within* the medium-term of 70 trading days, equivalent to about 3.5 months.

The 'meaningful rise target' for each index was established by considering 25% of the standard deviation relative to the index mean, calculated from the training periods pertinent to each index. This approach aims to set a significant yet attainable threshold over a medium-term period, reflecting the index's inherent volatility. Adjustments to these thresholds were empirically driven based on the training data to ensure a practical balance between predictive accuracy and a sufficient occurrence of the '1' label, thereby aligning with typical market behaviors over the long term.

For the NASDAQ 100, the target was set at 10%, for the DOW JONES at 8%, and for the DAX at 6%. The classification model uses two labels: label 1 is assigned to states where the index reaches the *target* at least once during the 70 trading days and does not drop more than one-third of the target value in the opposite direction. In other words, any *close* at time *t*, denoted as $close_t$, will have a label of 1 if the following condition is satisfied:

$$\begin{aligned}
&\exists i \in \overline{1..70} : close_{t+i}(close_t \cdot (1 + \frac{target}{100}) \leq close_{t+i}) \\
&\wedge \forall i \in \overline{1..70} : close_{t+i}(close_t \cdot (1 - \frac{target}{3 \cdot 100}) \leq close_{t+i})
\end{aligned} \tag{7}$$

In the classification model, we are looking for bottoms, but not too strictly like exact bottoms, since such points are rare and thus more difficult to classify. Conversely, label 0 is assigned to states where the above condition is not met. We use this binary classification to predict whether the index will experience a significant increase in price in the medium term.

B. FEATURE GENERATION AND NORMALIZATION

For each main index and its list of stocks, we generated all the indicators mentioned above. In some cases, there were companies that were initially traded after other stocks on the list for a given period of time. In such cases, the train set was set to start whenever there was complete data for all stocks involved. All the indicators created were then transformed by scaling each feature to a given range of [0, 1] using the Python MinMaxScaler. Normalizing the data is important because not all features are in the same units or magnitude, and scaling can help increase the accuracy of the models [24], [25]. Finally, all the normalized indicators of the main index and its list of stocks were set to be the input features for our classification model.

C. AI MODEL COMPARISON

We used 4 different classifiers and compared their performance effectiveness on the model. In this section, we will briefly review the classifiers used.

1) K-NEAREST NEIGHBORS CLASSIFIER

The K-nearest neighbors (KNN) algorithm is commonly used for classification and has been extensively applied in predicting stocks and indices [9], [26], [27]. The assumption behind KNN is that similar points can be found near one another. A class label is assigned based on a majority vote. For each single vector of input features in the test dataset, we look for the K vectors of input features in the train dataset that have the shortest distance from the test vector, among all other vectors in the train dataset. The label with the majority of the K vectors is set to be the classification label for the test vector.

To determine the distance, several methods can be used, with the most commonly used being the Euclidean distance between vectors, defined as

$$d(u, v) = \sqrt{\sum\nolimits_{i=1}^{n} (v_i - u_i)^2}$$

where u and v are vectors in the with n-dimensional space. In our study, we used the Python KNeighborsClassifier [28] with standard Euclidean distance, and K was usually set to 7, as this value showed the best performance.

2) RANDOM FOREST CLASSIFIER

The Random Forest Classifier (RF) is another important tool for predicting stocks and indices [29], [30], [31]. This classifier works by utilizing a collection of decision trees, each trained on a distinct part of the data, known as bootstrap sampling. Feature bagging is also applied, where different feature subsets enhance the model's robustness. Our method focuses on continuous indicators and their relationship, where this classifier analyzes them for the most effective division point and features on each level of the tree for achieving optimal node purity. We use the Python RandomForestClassifier [32] for our implementation. To measure the quality of a split in each tree, we use the default parameter *criterion*, which is set to Gini impurity. Gini impurity is a measure of the impurity of the nodes in the decision tree. It is calculated as $Gini = 1 - \sum_{i=1}^{n} P_i^2$, where n is the number of classes, and P_i is the probability of a data point being classified as class i. Regarding the number of trees in the forest, we use a value of 30, which has shown better performance on average than the default parameter of 100.

3) SVM CLASSIFIER

SVM has been extensively used for predicting stocks and indices [9], [26], [27]. It is particularly well-suited for binary classification when the data is non-linearly separable. The SVM algorithm operates by locating a hyperplane that maximizes the separation between the data points of distinct classes. Its primary objective is to enlarge the margin between the samples and the boundary delineating the classes. When the separation is complex and not linear, the input features are transformed into a higher-dimensional space, which makes it easier to create a linear boundary. This is done by using the kernel trick to enlarge the feature space. There are many different kernel functions that can be used, depending on the nature of the data.

In our research, we used the Python Support Vector Classification (SVC) [33]. Since we deal with non-linearly separable data, the kernel function of RBF was used. The tuning parameters resulted in C, the regularization parameter, being set to 200, and *gamma*, the kernel coefficient in the RBF function, being set to 1.0 after checking values from 0.01 to 10. The *gamma* coefficient should be positive, and a smaller value of *gamma* will lead to a decision boundary that is less sensitive to individual data points, and vice versa.

4) VOTING CLASSIFIER

The Voting Classifier is a commonly used ensemble method for stock predictions that combines the predictions of multiple classifiers to improve the accuracy and robustness of the model [34], [35], [36]. In our study, we used the Python implementation of the VotingClassifier from scikit-learn. The voting parameter is set to "soft", which means the prediction is based on the class probabilities rather than a simple majority vote.

D. ASSESSMENT OF CLASSIFICATION QUALITY

The assessment of classification quality is crucial in evaluating the performance of the models used. In this study, the *F1-score* was used to assess the precision and recall for the positive label (i.e., the '1' label). The precision and recall are equally important, but their significance depends on various factors such as the investor's trade preferences and the amount of resources available. For instance, if an investor has limited resources, she/he might prioritize precision over recall to minimize the risk of making mistakes. Conversely, if the investor has more resources and can afford to make more trades, she/he might prioritize recall over precision to maximize the number of profitable opportunities.

Furthermore, a high *F1-score* for the negative label (i.e., the '0' label) can give the investor useful hints when it's better NOT to enter the market, or in other words, when the probability of the index rising beyond the defined threshold is low.

In addition, the Cohen kappa score was employed to evaluate and compare different models in the presence of class imbalance. The Cohen kappa score is particularly useful as it adjusts for the probability of random agreement, providing a more robust indicator of model performance when classes are unevenly distributed.

VI. EXPERIMENTAL SETUP

For the three stock indices mentioned earlier, we tried different approaches and evaluation methods, as follows:

1) Using most correlated stocks with the index.
2) Using most weighted stocks that make up the index.
3) Using stock prices with generally accepted indicators.
4) Testing on different time periods.

TABLE 7. Dummy classifier, test results. (The last 350 days between January 1, 2010 and September 15, 2022).

Ticker	Strategy	Label	precision	recall	f1-score	accuracy	auc
^NDX	most frequent	0	0.82	1	0.9		
		1	0.0	0.0	0.0		
		all				0.82	0.5
^NDX	stratified	0	0.72	0.74	0.73		
		1	0.26	0.25	0.26		
		all				0.6	0.49
^DJI	most frequent	0	0.91	1.0	0.96		
		1	0.0	0.0	0.0		
		all				0.91	0.5
^DJI	stratified	0	0.9	0.74	0.81		
		1	0.06	0.17	0.08		
		all				0.69	0.45
^GDAXI	most frequent	0	0.83	1.0	0.91		
		1	0.0	0.0	0.0		
		all				0.83	0.5
^GDAXI	stratified	0	0.85	0.73	0.79		
		1	0.2	0.33	0.25		
		all				0.67	0.53

5) Determining under what circumstances the model can be utilized.
6) Testing the Selected Model Using a Rolling Forecast Approach.

In the following sections, we will detail the experiments conducted for each of the approaches, as well as the results obtained. It is important to note that we are dealing with an unbalanced class weight, since the label of 1 is much less frequent than the label of 0. This is because most of the time, situations where the index reaches the 'meaningful rise target' (label 1) are relatively infrequent compared to instances where it doesn't (label 0). The occurrence of label 1 cases varies, ranging from approximately 3% to 17% of the test data, with an average of around 10%. However, there are periods when these situations become notably rare, particularly in instances where the market experiences sustained downward trends. Recognizing these rare label 1 instances presents a significant challenge in our study, as their infrequent occurrence demands precise classification. We account for this class imbalance by setting the *class_weight* parameter of the classifiers accordingly, ensuring that our model is able to handle both the prevalent label 0 instances and the less frequent label 1 instances effectively. The *class_weight* adjustment aims to prevent the model from being overly biased towards the majority class and to enhance its ability to identify the rarer but significant upward movements that correspond to label 1.

Support Vector Machines (SVMs) are particularly sensitive to class imbalances, which can skew the hyperplane towards the majority class, thereby impairing the classifier's performance. As noted by Batuwita and Palade [37], SVM classifiers trained on imbalanced datasets tend to produce models biased towards the majority class, resulting in suboptimal performance on the minority class. They discuss various strategies to mitigate this imbalance, including adjusting class weights to compensate for disparities in class frequencies. This is addressed in our SVM model by applying the class_weight = 'balanced' setting.

Since the classes are imbalanced, the touchstone will be done by a comparison with the results obtained by a "dummy classifier," which will serve as a simple baseline. The dummy classifier will be activated in two different strategies: The first is by classifying all labels in the test set as the same as the *most frequent* label found in the train set. The second strategy is known as known as *stratified*, in which the classifier randomly selects labels according to the distribution of labels in the training set. This means that the ratio of each label's occurrence in the randomly generated predictions will be proportional to the ratio of that label's occurrence in the training set. Table 7 shows the results of the dummy classifier on the various indices and strategies.

VII. EXPERIMENTS AND RESULTS

A. MOST CORRELATED STOCKS WITH THE INDEX

Starting with NASDAQ-100, adding stocks in order according to Table 2, yielded the best results when adding only the first two stocks, i.e, Microsoft and Apple. This phenomenon repeated itself with all of the above classifiers. A trial in which the stock features were given more weight according to the stocks' relative weights in the index has not yielded better results, either. The best classifier was found to be the SVM in this case. For the Dow Jones, we acted according to Table 4, and added stocks one by one each time. The experiment reveals that the correlation approach is yielding poor results in all cases. This experiment was repeated with the DAX, according to Table 6 and as described before. The results are similar to those of NASDAQ-100 in that even here, using the first two stocks gave the best results, again using SVM. Table 8 summarizes the best results from the above experiments for the test set.

TABLE 8. Chosen stocks, most correlated approach, test results. (The last 350 days between January 1, 2010 and September 15, 2022).

Ticker	Best Classifier	Label	precision	recall	f1-score	accuracy	auc	Ticker used
^NDX	SVM	0	0.94	0.89	0.92			’MSFT’, ’AAPL’
		1	0.60	0.74	0.66			
		all				0.87	0.82	
^DJI	SVM	0	0.91	1.00	0.96			’HON’, ’AXP’, ’GS’, ’MSFT’
		1	0.00	0.00	0.00			
		all				0.91	0.50	
^GDAXI	SVM	0	0.9	0.83	0.86			’SIE.DE, ’ALV.DE’
		1	0.39	0.55	0.46			
		all				0.78	0.69	

TABLE 9. Chosen stocks, most weighted stocks approach, test results. (The last 350 days between January 1, 2010 and September 15, 2022).

Ticker	Best Classifier	Label	precision	recall	f1-score	accuracy	auc	Ticker used
^NDX	SVM	0	0.94	0.89	0.92			’AAPL’, ’MSFT’
		1	0.60	0.74	0.66			
		all				0.87	0.82	
^DJI	KNN	0	0.99	0.97	0.98			’UNH’, ’GS’, ’HD’
		1	0.70	0.87	0.78			
		all				0.96	0.92	
^GDAXI	Voting Classifier	0	0.94	0.79	0.86			’LIN.DE’, ’SAP.DE’ ’SIE.DE’
		1	0.42	0.74	0.53			
		all				0.79	0.77	

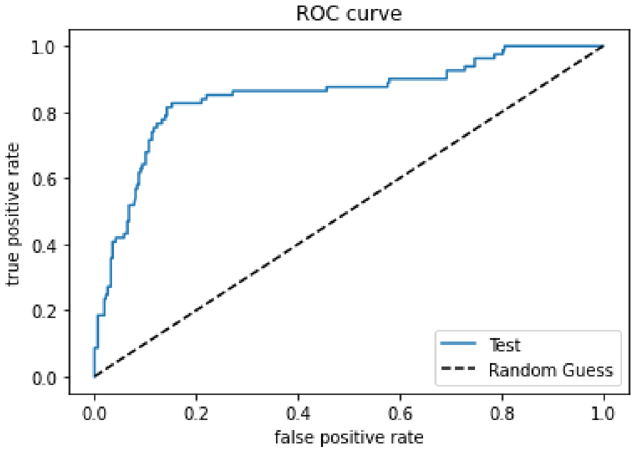

FIGURE 1. ROC curve for NASDAQ 100, using most weighted stocks that make up the index. (Test set results, the last 350 days between January 1, 2010 and September 15, 2022).

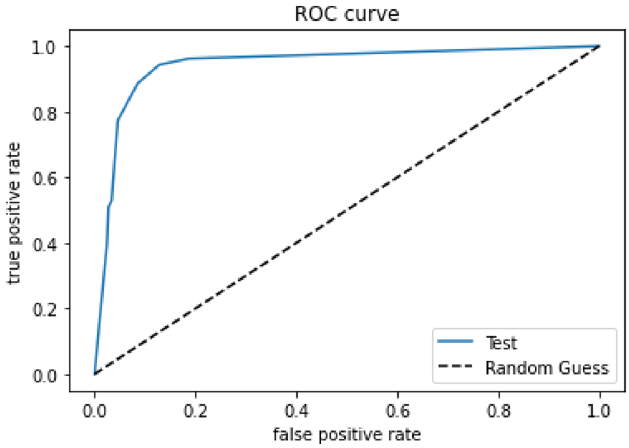

FIGURE 2. ROC curve for DOW, using most weighted stocks that make up the index. (Test set results, the last 350 days between January 1, 2010 and September 15, 2022).

B. MOST WEIGHTED STOCKS THAT MAKE UP THE INDEX

As previously mentioned, this approach involves adding stocks to the training set in descending order of their weight in the index. For NASDAQ 100, stocks were added according to Table 1. The best results were obtained by including only the two largest stocks, Apple and Microsoft, using the SVM classifier. Note that similar results were obtained by including the third stock, Amazon, and using the Voting Classifier.

For the Dow Jones, stocks were gradually selected according to Table 3. The best results were obtained by including only the first three stocks: United Health, Goldman Sachs, and Home Depot, using the KNN classifier. For the DAX, the best results were obtained by including the first three stocks: Linde, SAP, and Siemens, and using the Voting Classifier. It should be noted that the best cumulative weight for all indexes is around 25 percent, achieved by selecting only the first two to three stocks in the list.

We also observed that considering the relative weight of the input features did not contribute to predictability. Using the Voting Classifier yielded similar results to those obtained by the other classifiers.

Table 9 summarizes the best results from the above experiments for the test set.

Fig. 1, Fig. 2, and Fig. 3 show the ROC curves obtained for the three indexes. Fig. 4 and Fig. 5 show the classification results for this experiment on the charts of NASDAQ 100 and Dow Jones.

The test period (the orange line) is characterized by sudden ups and downs and with different lengths. The green circles

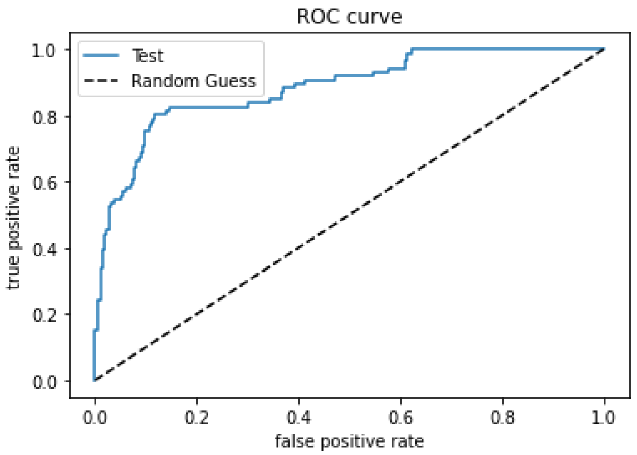

FIGURE 3. ROC curve for DAX, using most weighted stocks that make up the index. (Test set results, the last 350 days between January 1, 2010 and September 15, 2022).

represent successful classification, while the red circles represent unsuccessful ones. Although it is a stormy and volatile period, it can be seen that the classification succeeds many times. Moreover, we can see that even in the red circles, most of them will gain profits, which usually takes more than the defined 70 trading days to achieve, or sometimes we have to settle for more modest profits than what was defined. Furthermore, during the long descending period of the markets (around the last half of the test set), almost no circles are found, as if the classification feels that it is not the appropriate time to enter the markets.

C. COMPARISONS WITH STATE-OF-THE-ART METHODS

In our comprehensive analysis, we have explored a range of state-of-the-art machine learning models, including Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Logistic Regression.

Utilizing the MLPClassifier from sklearn [38], our ANN model integrates best practices and deep learning principles by Lecun et al. [39], along with recommended structures validated for efficacy in stock market prediction.

Our optimal ANN configuration utilized two hidden layers, with each layer sized at four times the number of input nodes, activation = 'relu' and solver = 'adam', for enhanced efficiency and speed. To mitigate overfitting, we set alpha = 0.01, complemented by a learning_rate_init of 0.001.

Our findings indicate that although ANNs deliver commendable results, they are slightly less favorable than the initially selected SVM model. For succinctness, we present the comparison for the NASDAQ 100 index; however, it is crucial to note that observations for other indices are analogous. The ANN achieved an F1-score for the '1' label of 0.62 and a Cohen Kappa score of 0.5, compared to the 0.66 F1-score and 0.58 Cohen Kappa achieved by the SVM model.

In addressing the challenges posed by a class imbalance within the context of time series data, we also considered using the Balanced Bagging Classifier from the imbalanced-learn library [40]. Given our data's nature and reliance on relative indicators to reflect the current market state, we proceeded with caution. The temporal integrity of our data is paramount. Thus, our primary evaluation focused on robust metrics such as Cohen's Kappa and the performance on the '1' label. Interestingly, integrating the BalancedBaggingClassifier did not substantially improve the outcomes and, in some instances, led to comparable or diminished performance.

Further emphasizing the SVM's effectiveness, we observed robust performance even without adjusting for class imbalances using the class_weight = 'balanced' setting. The Cohen's kappa showed minimal variation, underscoring the model's stability and reliability in handling unbalanced data. This consistent performance highlights the SVM's suitability for our dataset, where class imbalances are prevalent.

ANNs were chosen for their proficiency in modeling complex patterns, like financial time series. While initial comparisons reveal a marginally lower performance than SVMs, the flexible architecture of ANNs presents opportunities for further optimization. This makes it an appropriate classifier to be selected dynamically, where it might show the best result in other timeframes or indices.

Our CNN model builds on established methodologies, utilizing the Keras deep learning API [41] for optimal sequential data processing. Inspired by the work of Chen and He [42] and Hamoudi and Elseifi [43], we tailored our approach to harness the predictive power of CNNs for financial time series, with data organized into sequences using a window size of 5. The model incorporates two Conv1D layers with 128 filters each, designed to extract significant patterns from the data. To counteract overfitting, we integrated a 0.2 dropout rate. The architecture progresses to two Dense layers, concluding with a sigmoid activation for binary classification, demonstrating a balanced approach between model complexity and generalization capabilities. Despite its robust design, the CNN model did not surpass the performance of our previously selected classifiers. With the NASDAQ 100, it achieved an f1-score of 0.54 for the '1' label and Cohen's Kappa of 0.4, compared to the superior results of the SVM. Potential improvements could include experimenting with alternative architectures or integrating broader or different features.

In our exploration of LSTM models for stock price prediction, we utilized the Keras deep learning API [41], drawing upon a wealth of existing research that demonstrates the efficacy of LSTMs in this domain. The studies by Hamoudi and Elseifi [43], Moghar and Hamiche [44], and Gülmez [45] were instrumental in establishing the foundational insights that guided the architecture of our model. These works highlight the LSTM's capability to capture temporal dependencies in stock market data, a critical aspect we aimed to harness.

Our LSTM model was specifically designed with dual layers and integrated L2 regularization to address the challenges of overfitting while effectively managing the

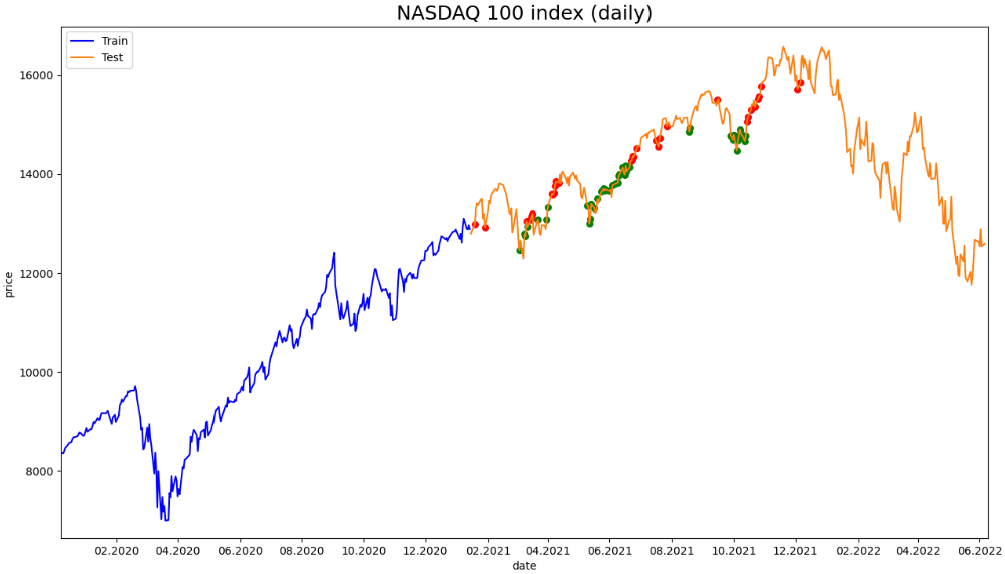

FIGURE 4. NASDAQ 100, Classification results. (Test results, the last 350 days between January 1, 2010 and September 15, 2022).

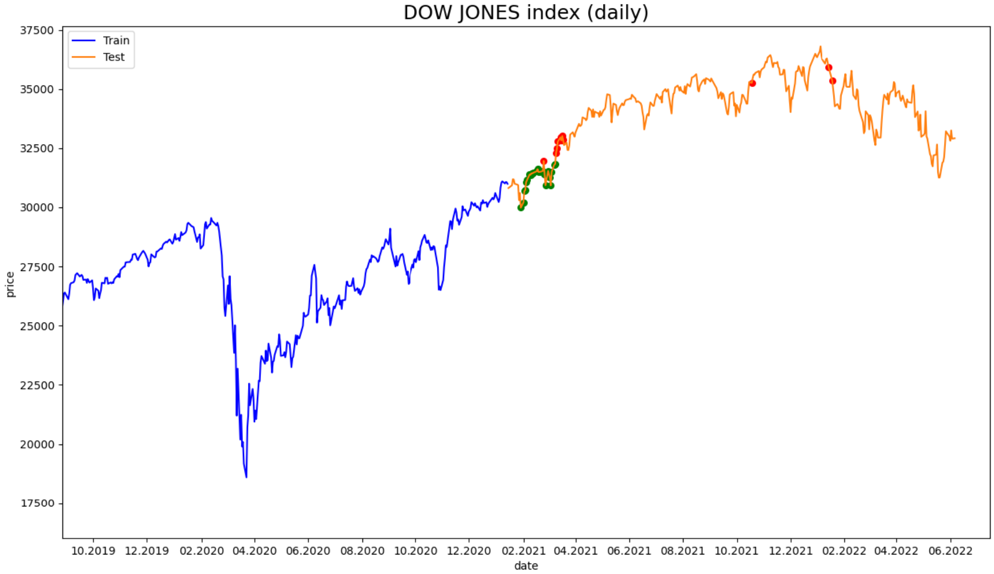

FIGURE 5. DOW JONES, Classification results. (Test results, the last 350 days between January 1, 2010 and September 15, 2022).

sequential nature of our dataset, which was organized into sequences using a window size of 5. Despite a rigorous process of hyperparameter tuning and the implementation of class weight adjustments to tackle class imbalance, our LSTM model's performance on the NASDAQ 100 index, marked by an f1-score of 0.4 and a Cohen's Kappa of 0.23 for the '1' label, did not meet the efficacy of our baseline SVM model. This observation was consistent with other indices examined in our study.

In addition to our primary analysis, which utilized SVM as our best-performing model, we integrated Logistic Regression to refine our baseline comparisons and further explore the landscape of machine learning approaches. Rigorously optimized to achieve the best possible F1-score for the '1' label, Logistic Regression's performance on the NASDAQ 100 index demonstrated notable improvements over simpler baseline models like the dummy classifier. However, it still lagged behind more complex models such as SVM and CNN. While the recall for the '1' label was high at 0.84, the precision was only 0.25, resulting in an F1-score of 0.38. Similar patterns were observed with the Dow and DAX indices, confirming the model's consistent performance limitations across diverse market conditions. This performance illustrates that Logistic Regression struggled to fully capture the complexity of market dynamics, primarily due to its linear nature, which is generally insufficient for modeling the nonlinear and dynamically changing relationships characteristic of stock market data.

1) RATIONALE BEHIND MODEL SELECTION

Our dynamic selection of SVM, KNN, Voting Classifier, and RF is driven by their proven performance, consistently outperforming more complex models like CNNs and ANNs in our dataset. SVM is valued for its ability to prevent overfitting and efficiently manage high-dimensional data, which is crucial for our study. KNN offers simplicity and effectiveness, capturing repeating patterns in the relative indicator values over time. The Voting Classifier integrates these models to enhance reliability and reduce bias. Moreover, CNNs and ANNs, while robust, require greater computational resources to achieve comparable results. The efficiency of SVM and KNN is particularly advantageous in our rolling forecast evaluations, which are inherently time-consuming, ensuring that our model selection is not only effective but also practical.

2) CHALLENGE OF DIRECT COMPARISONS

Our study's unique objective of predicting significant stock index rises over a defined medium-term renders direct comparisons with other studies challenging due to the diverse methodologies, classification definitions, and datasets employed across the financial machine-learning domain.

In comparing various machine learning models, we observed significant variability in methodologies and objectives.

Pagliaro [46] explored the effectiveness of the Extra Trees Classifier in predicting significant stock price movements over a short period of 10 trading days across 120 companies. He achieved an accuracy of 86%, highlighting the model's ability to handle large datasets enriched with numerous technical indicators. While his definition of 'significant,' tailored to short-term changes, is based on exceeding set percentage thresholds appropriate for a 10-day period, our study extends the forecasting horizon to the medium term. This differentiates our approach and provides a general benchmark for assessing the effectiveness of other state-of-the-art models in predicting significant market movements.

Milosevic [13] utilized advanced machine learning techniques, including RF, Logistic Regression, and SVM, to forecast significant long-term stock price rises, defined as a minimum 10% increase over one year. His study spanned 1,298 stocks, employing financial indicators like the Price-to-Earnings (P/E) ratio and Earnings Per Share (EPS) to label stocks that met this criterion as 'good' in a binary classification setup. He achieved a notable F1-score of 0.75, demonstrating robust predictive capability in a long-term investment context. This approach, on the one hand, provides a pertinent comparison to our study, but on the other hand, it differs significantly since our study concentrates on medium-term market movements, employing a distinct classification definition and a specific subset of stocks.

Mittal and Nagpal [14] developed a regression-based supervised learning model to predict stock returns over the medium to long term, defined as up to one year. They crafted a stock health index using a range of financial indicators, such as price-to-earnings ratios and earnings per share, coupled with fuzzy logic for actionable investment advice. This approach highlights their integration of advanced machine-learning techniques, providing a detailed performance validation with nearly 100% precision during a test period from May 3, 2020, to July 12, 2020. This period demonstrated substantial gains across all recommended stocks, validating the model's predictive accuracy and the reliability of the generated investment advice. While their study offers valuable insights into long-term investment strategies, it contrasts ours, concentrating on medium-term market movements with specific thresholds for significant financial events. This juxtaposition not only underscores the diversity within machine learning applications in financial forecasting but also enhances the comparative framework of our analysis, situating our methodology among varied approaches in the field.

D. STATISTICAL VALIDATION OF MODEL PERFORMANCE

To compare our selected models (SVM, KNN, Voting Classifier, RF) against baseline models (dummy classifiers) and more complex models (ANNs, CNNs, LSTM), we utilized the Mann-Whitney U test [47]. This non-parametric test, ideal for non-normally distributed data, assessed differences across multiple performance metrics, including Cohen's kappa and AUC, which are crucial given our dataset's imbalanced nature. The analyses were conducted across various experimental setups, including different timeframes and configurations. Our findings indicated significant performance enhancements with our primary models over the baselines (Mann-Whitney $U = 144.0$, $p = 1.46\text{e-}05$ for Cohen's kappa; Mann-Whitney $U = 144.0$, $p = 1.28\text{e-}05$ for AUC). Similar improvements were observed when compared to more complex models (ANNs, CNNs, LSTM), reinforcing the effectiveness and appropriateness of our model selection for predicting significant stock index rises.

E. USING STOCK PRICES WITH GENERALLY ACCEPTED INDICATORS

The majority of studies in the field of stock classification use prices themselves as input features, together with commonly used indicators in their customary form [9], [15], and [48]. Accordingly, we used the prices of the index and the stocks that make up the index, together with indicators in their customary form. Stocks were selected according to weights and, in another experiment, according to their correlation with the main index. After many trials, we found that using only one indicator, the simple moving average (SMA), yielded the best results. The SMA is generally accepted and is defined as follows:

$$SMA_t(n) = \frac{1}{n} \sum_{i=t-n+1}^{t} close_i \quad (8)$$

where n represents the time periods and t represents the data point at period t. We used three different parameters for the SMA: $SMA_t(5)$, $SMA_t(7)$, and $SMA_t(10)$. The input features thus contain the closing prices of the main index, the closing prices of the selected stocks, and the three *SMA*s with the above parameters.

Although this approach may seem unlikely on the face of it, since it suggests that the prices themselves have predictability without the patterns they exhibit, trying this approach on the NASDAQ 100 yielded the best results so far. The best results were obtained when the selected stocks were chosen according to their weights, with the first 10 stocks selected from Table 1, and the KNN classifier was used (see Table 10). However, trying the approach on the Dow and the DAX yielded only poor results, as expected, which were similar to those obtained by the dummy classifier and hence are not shown in Table 10.

F. TESTING ON DIFFERENT TIME PERIODS

At this stage, we tested the models on different time periods with the above indexes. The results reveal the following:

1) The best model is still the most weighted stocks with the newly created relative indicators
2) For different time periods, the best classifier might sometimes be changed for better performances
3) The model has good predictability in many different time periods with the above indexes
4) In many cases, the model has less favorable predictability, but it can still giving us useful information for better entry points, hence reduce trade risks. Consider, for example, the Dow between the time periods of January 01, 2000 to July 21, 2015. This is a time period with completely different economic data and different patterns in the markets. Table 11 shows the model performance using RF with a test size of 350 as before. It can be seen that the *F1-score* for the positive label (i.e., the 1 label) is only around 0.5. Although the results are less favorable than those achieved in the previous period, they can still give a strong hint when it's a better time to enter the market, or more than it, when NOT to enter. For the sake of comparison, the dummy classifier for this period of time gives a *F1-score* for the positive label around 0 to 0.1, depending on the strategy.
5) In some cases, the model demonstrates predictive capabilities that are similar to those demonstrated by the dummy classifier. This is expected because of the random nature of the markets. So, our study shows that many times the model is efficient, hence the markets are not completely random in certain periods of time, and on the other hand, in other periods of time the markets are indeed random and behave according to the Random Walk Theory [7]. Therefore, at this stage, the goal is to know in which time periods the model can be used and when not. For this purpose, we performed the experiment described in the next section.

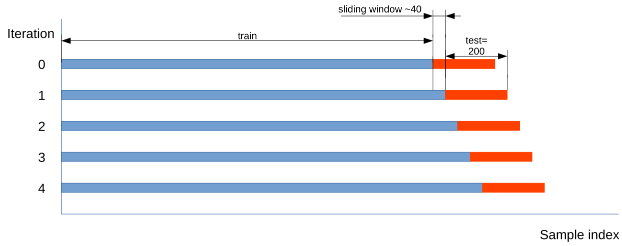

FIGURE 6. **Cross-validation of the model's predictability over time using a sliding window technique. Each line represents a different experiment, and the blue area denotes the train set while the red area denotes the test set.**

G. DETERMINE UNDER WHAT CIRCUMSTANCES THE MODEL CAN BE UTILIZED

The testing size we used, 350 days, actually represents a wide range of time, around a year and a half of trading in the markets. Our assumption is that when the model has acceptable predictability in such a wide period of time, we will be able to use it with similar capabilities also in the following short period of time, i.e., the near unknown future right after the test period. To investigate this assumption, we used different periods of time with different indexes using the model described above. The experiment was done using a technique similar to cross-validation of time series. This time, the test size was set to an even shorter period, 200 days (around 10 months), and represents the last known completed classification test set. The next two months from there (about 40 trading days) represent the unknown near future. Since the model is defined for 70 trading days ahead (around 3.5 months), and in addition to that, the positive label is relatively rare, it's not enough to check only two months ahead, so we moved the test period forward in a sliding window, each step of two months (about 40 days) ahead, and we rechecked the results of the new test period each time. As usual, when dealing with time series, no information was 'leaked' from the future to the past, since the window is sliding forward only. The goal of this experiment was to track the *auc score*, i.e., the area beneath the ROC curve. Fig. 6 depicts the process used to evaluate the *auc score* over time. It turns out that the predictability represented by the *auc* is usually changed gradually while the window is sliding, i.e., we can use the model for the next short period as long as the results of the last test were satisfactory enough for us. When the quality of the results starts to deteriorate, it is usually a good idea to temporarily leave the model and wait until the quality of the results will get improved above a threshold of randomness. These results were obtained for the various indexes aforementioned as well as for different periods of

TABLE 10. Using stock prices with generally accepted indicators, test results. (The last 350 days between January 1, 2010 and September 15, 2022).

Ticker	Best Classifier	Label	precision	recall	f1-score	accuracy	auc	Ticker used
^NDX	KNN	0	0.95	0.97	0.96			10 most weighted
		1	0.83	0.77	0.80			
		all				0.93	0.87	

TABLE 11. Dow, test results of the last 350 days between January 01, 2000 and July 21, 2015.

Ticker	Best Classifier	Label	precision	recall	f1-score	accuracy	auc	Ticker used
^DJI	RF	0	0.99	0.96	0.97			'UNH', 'GS', 'HD'
		1	0.42	0.77	0.54			
		all				0.95	0.86	

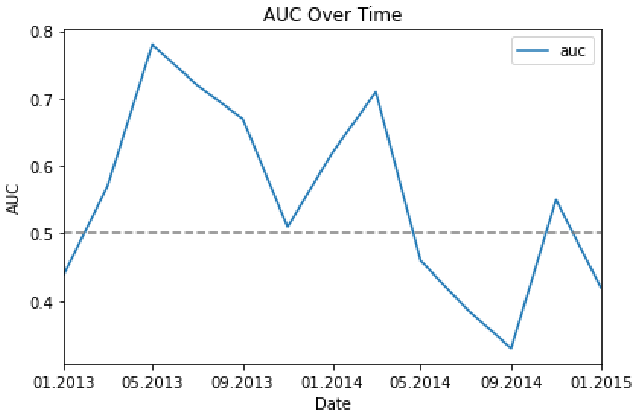

FIGURE 7. Dow, auc over time, January 01, 2013 to January 01, 2015.

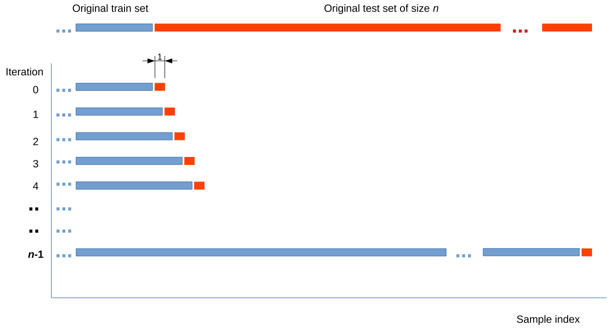

FIGURE 8. Rolling forecast process description.

time. As an explanation, it seems that the random nature of the markets takes place in the descent part of the time, but our goal is to identify periods when the market is less efficient; hence, we can use our model since those periods are usually built gradually and fade gradually. As an example, consider Fig. 7, which depicts the *auc score* of the Dow Jones during the period of time between January 01, 2013 to January 01, 2015. It can be seen that only after February 2013, the *auc score* signals predictability, where it is consistently above the 0.5 threshold of randomness. It is worth mentioning that this period lasted until April 2014, a significant period of time, which is more than a year of trading in the market.

Analysis of AUC trends reveals that periods characterized by relatively stable market volatility, alongside clear and long-lasting trends, correlate with higher predictive accuracy, where AUC consistently surpasses the 0.5 threshold. In contrast, during times of increased or unstable market volatility, AUC scores typically decline, indicating reduced model effectiveness. These observations suggest that the model is most effective during times when the market shows inefficiencies, offering potential profit opportunities. However, in highly efficient, randomly behaving markets, the model's predictive reliability diminishes.

Therefore, while the primary strategy for deploying the model involves closely monitoring the AUC scores, it is also beneficial to observe market volatility and prevailing trends. This combined approach enhances investment decisions by leveraging periods that are optimal for the model's capabilities, with AUC scores serving as the principal guide supported by additional market condition insights.

H. TESTING THE SELECTED MODEL USING A ROLLING FORECAST APPROACH

At this stage, the model will be tested using a rolling forecast approach, where the training set increases in size during each testing phase as it extends over an increasing range. This means that the model is tested for each new day in the original test set while the training set is updated to include all previously known classification information up to that day. This approach creates more accurate testing, as it leverages all the information already known and predicts only the next day each time. This process is repeated for the entire original test set, with a new testing process for each new day.

Fig. 8 depicts the rolling forecast process used. The original test set of size *n* is shown at the top, and the rolling process is depicted below. In previous stages of this research, the entire test set was evaluated as a whole after the model was trained. However, with the rolling forecast approach, only the next day in the test set is tested in each iteration after refitting the growing training set. In the end, the individual results of each day are combined into one result that represents the result of the entire original test set The

TABLE 12. Rolling forecast, most weighted stocks approach, test results. (The last 350 days between January 1, 2010 and September 15, 2022).

Ticker	Best Classifier	Label	precision	recall	f1-score	accuracy	auc	Ticker used
^NDX	SVM	0	0.98	0.97	0.98			'AAPL', 'MSFT'
		1	0.88	0.92	0.90			
		all				0.96	0.95	
^DJI	KNN	0	0.98	0.98	0.98			'UNH', 'GS', 'HD'
		1	0.83	0.83	0.83			
		all				0.97	0.91	
^GDAXI	Voting Classifier	0	0.97	0.98	0.98			'LIN.DE', 'SAP.DE' 'SIE.DE'
		1	0.91	0.84	0.88			
		all				0.96	0.91	

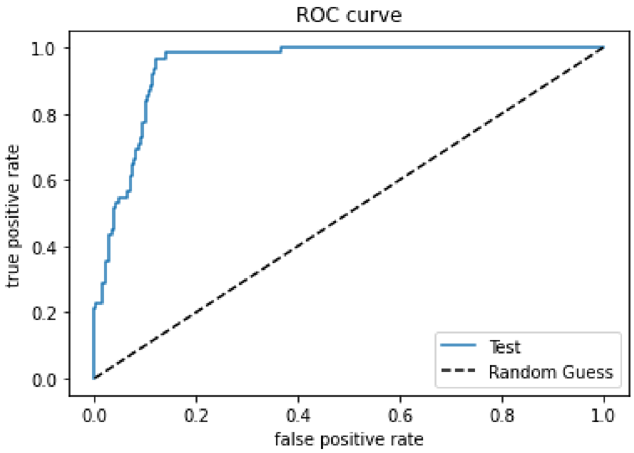

FIGURE 9. Rolling forecast. ROC curve for NASDAQ 100, using most weighted stocks that make up the index. (Test set results, the last 350 days between January 1, 2010 and September 15, 2022).

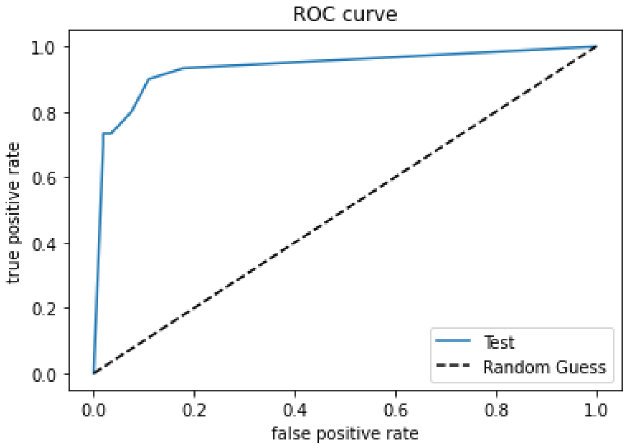

FIGURE 10. Rolling forecast. ROC curve for DOW, using most weighted stocks that make up the index. (Test set results, the last 350 days between January 1, 2010 and September 15, 2022).

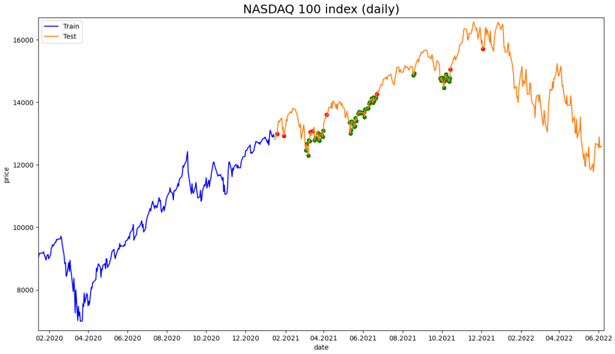

FIGURE 11. NASDAQ 100, Classification results by rolling forecast. (Test results, the last 350 days between January 1, 2010 and September 15, 2022).

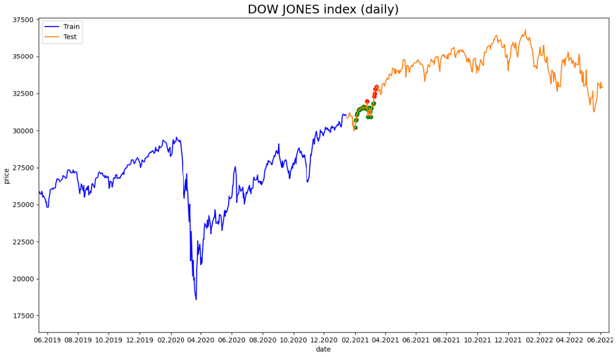

FIGURE 12. DOW JONES, Classification results by rolling forecast. (Test results, the last 350 days between January 1, 2010 and September 15, 2022).

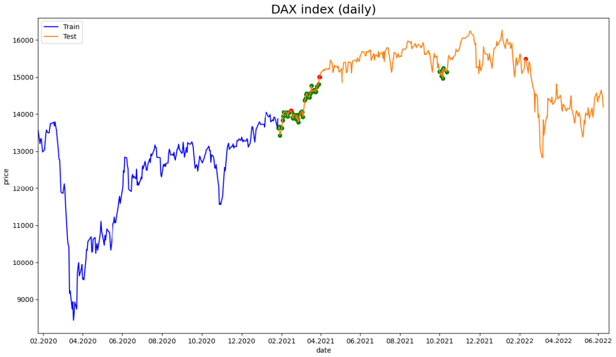

FIGURE 13. DAX, Classification results by rolling forecast. (Test results, the last 350 days between January 1, 2010 and September 15, 2022).

results show that, on average, this rolling forecast approach outperforms the previous approach. Consider the experiment shown in Table 9. Repeating this experiment using the rolling forecast approach yielded the results depicted in Table 12. Additionally, Fig. 9 and Fig. 10 depict the ROC curves obtained for NASDAQ 100 and DOW with these experiments. We can see that by using this method of rolling forecast, the results are outperforming the previous method, which refers to the test set as one whole set. For more comparison, Fig. 11 Fig. 12 show the classification results as circles on the NASDAQ 100 and the DOW charts. Fig. 4

and Fig. 5, as shown earlier, depict the same experiment results in an unrolling manner. The differences between the two pairs of charts are evident: the rolling forecast approach shows more green circles and fewer red ones, indicating an improvement in accuracy. In addition, Fig. 13 depicts the classification results for the same period of time for the DAX index, with the rolling forecast. Again, it can be seen that although it is a very turbulent period in that market, still there are almost no red circles at all, with a lot of successful green circles.

VIII. CONCLUSION AND FUTURE WORK

A. AN OVERVIEW OF THE BENEFITS AND CONTRIBUTIONS OF OUR RESEARCH AND MODEL

Overall, our research and model make several contributions to the field. Firstly, we address the issue of predicting stock index direction in the medium term, an area that has received little attention in the existing literature. Secondly, we introduce new relative indicators that are specifically designed to predict upward significant movements in the stock indexes. Thirdly, we demonstrate that selecting stocks based on their relative weight in the index yields better results than other selection criteria, such as correlation with the index.

Our experiments on three different indexes - the Dow Jones Industrial, NASDAQ 100, and the German DAX - show that our model can be very effective in predicting the positive label (i.e., upward significant movement) with an *F1-score* of up to 0.8 and an AUC of around 0.8 to 0.9. In addition, we demonstrate that our model can be useful even in periods of less predictable market behavior by providing clues as to when it is not a good time to enter the market or when significant moves are not likely to happen.

Furthermore, we show that the choice of classifier should be tailored to each period and index, although some classifiers may be more fitting for certain indexes. Our research also supports the random walk theory in that predictability can be extremely difficult in decent parts of the time. However, we demonstrate that following the AUC over time with our proposed model can give us insight into when the market deviates from its typical random path, and predictability becomes possible.

Finally, we show that our model's performance can be improved by using a rolling forecast method to test the data. By dividing the test set into single days and refitting the training set for each day, we obtain more accurate results that outperform the common testing procedure. By implementing the rolling forecast in conjunction with the most weighted stocks approach, previously demonstrated as highly predictable, our model demonstrates predictive accuracy for the positive label, achieving *F1-score* as high as 0.9 and AUC values ranging from 0.9 to 0.95 across various indices. Importantly, our study introduces a novel methodology for medium-term stock index prediction that can be practically applied by investors of all types looking to enhance their risk/reward ratio.

B. FUTURE RESEARCH DIRECTIONS

Looking ahead, our research opens the door to several exciting opportunities. A primary direction involves applying our model to a wider range of stock indices and the individual stocks comprising those indices since, as previously described, our investigation was conducted on a select few but major global indices. This extension could yield deeper insights and potential enhancements to the model. Experimenting with alternative classifiers and diverse combinations of relative indicators could further enhance the model's predictive strength. An additional intriguing area to explore is investors' sentiment within indices and their constituent stocks. This exploration could include developing a new indicator to analyze shifts in investor sentiment, independent of price movements, serving as an additional feature to enrich the model.

ACKNOWLEDGMENT

This research was undertaken with the guidance and support of Prof. Dr. B. Pârv from the Faculty of Mathematics and Computer Science. A. Bareket extend his sincere thanks to Prof. B. Pârv for his invaluable mentorship and expertise, which significantly contributed to this work.

REFERENCES

[1] D. Rey, "Stock market predictability: Is it there? A critical review," WWZ/Dept. Finance, Univ. Basel, Basel, Switzerland, Work. Paper 12/03, 2004.

[2] X. Zhong and D. Enke, "Forecasting daily stock market return using dimensionality reduction," *Exp. Syst. Appl.*, vol. 67, pp. 126–139, Jan. 2017, doi: 10.1016/j.eswa.2016.09.027.

[3] L. D. Persio and O. Honchar, "Artificial neural networks architectures for stock price prediction: Comparisons and applications," Dept. Comput. Sci., Univ. Verona, Verona, Italy, Tech. Rep., 2016.

[4] A. Zheng and J. Jin, "Using AI to make predictions on stock market," Stanford Univ., Stanford, CA, USA, Tech. Rep., 2018.

[5] E. F. Fama, "Random walks in stock market prices," *Financial Analysts J.*, vol. 51, no. 1, pp. 75–80, 1995, doi: 10.2469/faj.v51.n1.1861.

[6] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *J. Finance*, vol. 25, no. 2, p. 383, May 1970.

[7] B. G. Malkiel, *A Random Walk Down Wall Street The Time-Tested Strategy for Successful Investing*. New York, NY, USA: WW Norton Company, 2021.

[8] N. Keimling, "Predicting stock market returns using the shiller CAPE—An improvement towards traditional value indicators?" Taunus Trust GmbH, 2016. [Online]. Available: http://ssrn.com/abstract=2736423http://www.starcapital.de/research/stockmarketvaluation.Electroniccopyavailableat:https://ssrn.com/abstract=2736423

[9] Y. Chen and Y. Hao, "A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction," *Exp. Syst. Appl.*, vol. 80, pp. 340–355, Sep. 2017, doi: 10.1016/j.eswa.2017.02.044.

[10] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS ONE*, vol. 12, no. 7, Jul. 2017, Art. no. e0180944, doi: 10.1371/journal.pone.0180944.

[11] R. Akita, A. Yoshihara, T. Matsubara, and K. Uehara, "Deep learning for stock prediction using numerical and textual information," in *Proc. IEEE/ACIS 15th Int. Conf. Comput. Inf. Sci. (ICIS)*, Jun. 2016, pp. 1–6, doi: 10.1109/ICIS.2016.7550882.

[12] H. Hu, L. Tang, S. Zhang, and H. Wang, "Predicting the direction of stock markets using optimized neural networks with Google Trends," *Neurocomputing*, vol. 285, pp. 188–195, Apr. 2018, doi: 10.1016/j.neucom.2018.01.038.

[13] N. Milosevic, "Equity forecast: Predicting long term stock price movement using machine learning," 2016, *arXiv:1603.00751*.

[14] S. Mittal and C. K. Nagpal, ''Predicting a reliable stock for mid and long term investment,'' *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 10, pp. 8440–8448, Nov. 2022, doi: 10.1016/j.jksuci.2021.08.022.

[15] M. Qiu and Y. Song, ''Predicting the direction of stock market index movement using an optimized artificial neural network model,'' *PLoS ONE*, vol. 11, no. 5, May 2016, Art. no. e0155133, doi: 10.1371/journal.pone.0155133.

[16] L. Lei, ''Wavelet neural network prediction method of stock price trend based on rough set attribute reduction,'' *Appl. Soft Comput.*, vol. 62, pp. 923–932, Jan. 2018, doi: 10.1016/j.asoc.2017.09.029.

[17] (2024). *Yahoo Finance*. [Online]. Available: https://finance.yahoo.com/

[18] (2024). *Yfinance Python Package*. [Online]. Available: https://pypi.org/project/yfinance/

[19] (2024). *Sklearn.metrics: Metrics and Scoring: Quantifying the Quality of Predictions*. [Online]. Available: https://scikit-learn.org/stable/modules/modelevaluation.html

[20] C.-H. Park and S. H. Irwin, ''The profitability of technical analysis: A review,'' AgMAS Project, Res. Rep. 2004-04, 2004.

[21] R. Cervelló-Royo, F. Guijarro, and K. Michniuk, ''Stock market trading rule based on pattern recognition and technical analysis: Forecasting the DJIA index with intraday data,'' *Exp. Syst. Appl.*, vol. 42, no. 14, pp. 5963–5975, Aug. 2015, doi: 10.1016/j.eswa.2015.03.017.

[22] S. Agrawal, D. A. U. Khan, and D. P. K. Shukla, ''Stock price prediction using technical indicators: A predictive model using optimal deep learning,'' *Int. J. Recent Technol. Eng. (IJRTE)*, vol. 8, no. 2, pp. 2297–2305, Jul. 2019, doi: 10.35940/ijrteb3048.078219.

[23] X. Di, ''Stock trend prediction with technical indicators using SVM,'' Stanford Univ., Stanford, CA, USA, Tech. Rep., 2014. [Online]. Available: http://finance.yahoo.com

[24] D. Singh and B. Singh, ''Investigating the impact of data normalization on classification performance,'' *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 105524, doi: 10.1016/j.asoc.2019.105524.

[25] L. B. V. de Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, ''The choice of scaling technique matters for classification performance,'' *Appl. Soft Comput.*, vol. 133, Jan. 2023, Art. no. 109924, doi: 10.1016/j.asoc.2022.109924.

[26] D. A. Puspitasari and Z. Rustam, ''Application of SVM-KNN using SVR as feature selection on stock analysis for Indonesia stock exchange,'' in *Proc. AIP Conf.*, 2018, Art. no. 020207, doi: 10.1063/1.5064204.

[27] R. K. Nayak, D. Mishra, and A. K. Rath, ''A Naïve SVM-KNN based stock market trend reversal analysis for Indian benchmark indices,'' *Appl. Soft Comput.*, vol. 35, pp. 670–680, Oct. 2015, doi: 10.1016/j.asoc.2015.06.040.

[28] (2024). *Sklearn.neighbors.KNeighborsClassifier*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[29] C. Lohrmann and P. Luukka, ''Classification of intraday S&P500 returns with a random forest,'' *Int. J. Forecasting*, vol. 35, no. 1, pp. 390–407, Jan. 2019, doi: 10.1016/j.ijforecast.2018.08.004.

[30] L. Khaidem, S. Saha, and S. Roy Dey, ''Predicting the direction of stock market prices using random forest,'' 2016, *arXiv:1605.00003*.

[31] S. Basak, S. Kar, S. Saha, L. Khaidem, and S. R. Dey, ''Predicting the direction of stock market prices using tree-based classifiers,'' *North Amer. J. Econ. Finance*, vol. 47, pp. 552–567, Jan. 2019, doi: 10.1016/j.najef.2018.06.013.

[32] (2024). *Sklearn.ensemble.RandomForestClassifier*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[33] (2024). *Sklearn.SVM.SVC*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.SVM.SVC.html

[34] C.-H. Chen, P.-Y. Chen, and J. Chun-Wei Lin, ''An ensemble classifier for stock trend prediction using sentence-level Chinese news sentiment and technical indicators,'' *Int. J. Interact. Multimedia Artif. Intell.*, vol. 7, no. 3, p. 53, 2022, doi: 10.9781/ijimai.2022.02.004.

[35] B. Qian and K. Rasheed, ''Stock market prediction with multiple classifiers,'' *Int. J. Speech Technol.*, vol. 26, no. 1, pp. 25–33, Jan. 2007, doi: 10.1007/s10489-006-0001-7.

[36] M. Rashidpoor Toochaei and F. Moeini, ''Evaluating the performance of ensemble classifiers in stock returns prediction using effective features,'' *Exp. Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 119186, doi: 10.1016/j.eswa.2022.119186.

[37] R. Batuwita and V. Palade, ''Class imbalance learning methods for support vector machines,'' in *Imbalanced Learning: Foundations, Algorithms, and Applications*. Hoboken, NJ, USA: Wiley, 2013, pp. 83–99.

[38] (2024). *Sklearn.neuralnetwork.MLPClassifier*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.neuralnetwork.MLPClassifier.html

[39] Y. Lecun, Y. Bengio, and G. Hinton, ''Deep learning,'' *Nature*, vol. 521, no. 7553, pp. 436–444, 7553. [Online]. Available: http://colah.github.io/

[40] (2024). *BalancedBaggingClassifier*. [Online]. Available: https://imbalanced-learn.org/stable/references/generated/imblearn.ensemble.BalancedBaggingClassifier.html

[41] (2024). *Keras*. [Online]. Available: https://keras.io/

[42] S. Chen and H. He, ''Stock prediction using convolutional neural network,'' *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 435, no. 1, Nov. 2018, Art. no. 012026, doi: 10.1088/1757-899X/435/1/012026.

[43] H. Hamoudi and M. A. Elseifi, ''Stock market prediction using CNN and LSTM,'' Comput. Sci. Dept., Stanford Univ., Stanford, CA, USA, Tech. Rep., 2021.

[44] A. Moghar and M. Hamiche, ''Stock market prediction using LSTM recurrent neural network,'' *Proc. Comput. Sci.*, vol. 170, pp. 1168–1173, Jan. 2020, doi: 10.1016/j.procs.2020.03.049.

[45] B. Gülmez, ''Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm,'' *Exp. Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120346, doi: 10.1016/j.eswa.2023.120346.

[46] A. Pagliaro, ''Forecasting significant stock market price changes using machine learning: Extra trees classifier leads,'' *Electronics*, vol. 12, no. 21, p. 4551, Nov. 2023, doi: 10.3390/electronics12214551.

[47] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric Statistical Methods* (Wiley Series in Probability and Statistics). Hoboken, NJ, USA: Wiley, 2013. [Online]. Available: https://books.google.co.il/books?id=Y5s3AgAAQBAJ

[48] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, ''Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques,'' *Exp. Syst. Appl.*, vol. 42, no. 1, pp. 259–268, Jan. 2015, doi: 10.1016/j.eswa.2014.07.040.

A. BAREKET received the B.A. degree (magna cum laude) in economics from Tel Aviv University, and the B.Sc. degree in computer science and the M.B.A. degree from Bar Ilan University. He has established a solid academic foundation in Israel. His diverse career spanning high-tech, finance, and the academic world has provided a rich foundation for innovative approaches. This background naturally led him to Ph.D. studies in computer science, with a primary research focus on applying AI to predict stock indices, leveraging various algorithms and methods in machine learning. His academic and professional journey is marked by a dedication to bridging theoretical knowledge with practical solutions, particularly in computer science with an emphasis on programming languages, and machine learning.

B. PÂRV is currently a Professor Emeritus with the Faculty of Mathematics and Computer Science, Department of Computer Science, Babeş-Bolyai University, Cluj-Napoca, Romania. With a distinguished career in academia, he has made significant contributions to the field of computer science, particularly in computational methods and their applications in diverse research areas. His expertise spans a broad range of topics within computer science, and he is known for his dedication to advancing the field through both research and teaching. His work has been instrumental in shaping the understanding and application of computer science principles in various complex scenarios.