Machine Learning with Applications 20 (2025) 100631

Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Key technical indicators for stock market prediction

Seyed Mostafa Mostafavi [a,*], Ali Reza Hooman [b]

[a] Imperial College London, London, United Kingdom
[b] SMM Trading Services, London, United Kingdom

ARTICLE INFO

Keywords:
Technical indicator
Stock market
S&P 500 index
Machine learning
LSTM
XGBoost
Random forest
SVR
PCA

ABSTRACT

The use of technical indicators for forecasting the stock market is widespread among investors and researchers. It is crucial to determine the optimal number of input technical indicators to predict the stock market successfully. However, there is no consensus on which collection of technical indicators is most suitable. The selection of technical indicators for a given forecasting model continues to be an active area of research. To our knowledge, there is limited published work on the importance of technical indicators in various categories such as momentum, trend, volatility, and volume. To identify the key technical indicators for stock market prediction, we employed XGBoost, Random Forest, Support Vector Regression, and LSTM regression techniques using 88 technical indicators as input data. We also used the PCA method for dimension reduction. The results reveal the most significant technical indicators within the momentum, trend, volatility, and volume categories. Our findings provide evidence that the proposed model is highly effective in predicting daily prices (with and without lag in Close price) on the S&P 500 stock index.

1. Introduction

The S&P 500 is a market capitalization index that tracks the daily activities of America's largest companies. By doing so, it reflects the overall performance of the US stock market. As the US economy is represented by this index, its movements have a significant impact on global financial markets due to the size and interactions of these companies. Moreover, the S&P 500 Index is considered one of the leading indicators of global financial health. As a result, investors and traders from around the world closely monitor its movements (Phillips & Shi, 2020).

Different approaches have been used to forecast market patterns. These include fundamental and technical analysis, statistical and multi-criteria decision-making, text analysis, data mining, and soft computing. Machine learning (ML) algorithms, a subset of statistical methods, are gaining popularity in the financial industry (Board, 2017).

Machine learning (ML) has the potential to identify stock trends and uncover underlying stock price dynamics by analyzing vast amounts of data. In computational finance, predicting financial markets through ML techniques often involves using technical indicators as attributes or features in the input datasets (Fernández et al., 2023). However, determining which technical indicators are suitable for a specific forecasting model is still an area of active research. Researchers vary in their use of technical indicators, with some utilizing ten, while others employ twenty or more (Alsubaie, Hindi, and Alsalman, 2019). There are already numerous technical indicators used by investors, and this number continues to grow. (Basak et al., 2019; Chen and Hao, 2017; Qiu and Song, 2016; Song, Lee, and Lee, 2019; Weng et al., 2018).

The authors (Gorenc Novak & Velušček, 2016) conducted a study on predicting the daily high for 370 S&P 500 companies from 2004 to 2013. They applied the Guided D-threshold strategies using 13 technical indicators, both with and without statistical classifiers including Support Vector Machine (SVM), Linear Discriminant Analysis (LDA), and Naïve Classifier. After analyzing the results, it was found that the support vector machine classifier with a Gaussian RBF kernel performed the best in terms of classification results and overall performance on the Guided D-threshold strategy. To train their model, the researchers used a 500-day rolling window period. Additionally, they performed a grid search to find the optimal model refresh period. The options considered were 5, 10, 20, 40, 80, and 160 days. It was found that a refresh period of 20 days achieved the highest accuracy.

Recent advancements in machine learning (ML) have significantly advanced the field of financial forecasting, offering transformative potential for predicting stock market dynamics. Fernández et al. (2023) conducted a pivotal study demonstrating that integrating technical indicators (e.g., moving average convergence divergence, relative

[*] Corresponding author.
E-mail addresses: smm98@ic.ac.uk (S.M. Mostafavi), alireza.hooman@smmtradingservices.com (A.R. Hooman).

https://doi.org/10.1016/j.mlwa.2025.100631
Received 25 October 2024; Received in revised form 10 February 2025; Accepted 16 February 2025
Available online 1 March 2025

strength index) into ML frameworks reduced prediction error by 32% relative to traditional autoregressive integrated moving average (ARIMA) models. This finding underscores the critical role of supplementary variables in enhancing model accuracy, particularly in capturing nonlinear market behaviors that linear statistical methods fail to address.

Complementing this work, Ampomah et al. (2021) proposed a hybrid forecasting framework combining machine learning algorithms with tree-based ensemble techniques, such as gradient-boosted decision trees. Their model achieved superior performance compared to conventional moving average strategies, particularly during periods of high market volatility. The study highlighted the capacity of ML-driven approaches to adapt to abrupt shifts in market regimes, a capability absent in classical technical indicators like Bollinger Bands, which rely on static volatility thresholds (Ampomah et al., 2021).

In their study, (Zhong & Hitchcock, 2021) introduced a method that combines various predictive models to forecast stock prices. This approach incorporates seven technical indicators, fundamental characteristics, and sentiment analysis based on text data. The authors achieved an accuracy rate of 66.18% for predicting S&P 500 index prices and 62.09% for individual stock prediction. To create their state-of-the-art ensemble models, they employed machine learning (ML) algorithms such as Random Forest and LSTM. The dataset used in this research consists of weekly historical prices spanning from January 1, 2000, to December 31, 2019.

In addition, it is crucial to use a minimal number of input features when forecasting stock market trends (Basak et al., 2019). Each input feature adds another dimension to the data, increasing its sparsity in the data space. As a result, the amount of training data needed grows exponentially as more features are included to cover every combination of feature values. Hence, it is important to identify the most significant technical indicators for predicting stock market trends. This not only enhances the accuracy of machine learning (ML) predictions but also saves time and reduces risks associated with stock market trading or investing. By focusing on the most essential indicators, traders can make more informed decisions about when to buy or sell stocks, ultimately improving the performance of their trading or investing strategies.

Therefore, our objective was to determine the crucial indicators for predicting stock market prices of the S&P 500, which serves as a pivotal benchmark for both the U.S. and global stock markets. To achieve this, we focused on four main categories of indicators: volume, volatility, trend, and momentum. We aimed to identify the most significant indicators in each category, providing valuable insights into profitable technical indicators that have proven effective in forecasting the S&P 500 stock index.

Moreover, according to the latest surveys and our extensive knowledge in the field, most of the research on market price prediction primarily revolves around classification techniques. Surprisingly, very few studies have delved into the realm of single or multiple regression approaches (Ayitey Junior et al., 2023; Obthong, 2020; Rundo et al., 2019). Consequently, we have carefully chosen the most vital and highest-performing multivariate regression techniques to yield optimal results in our predictive research. Given the inherent time dependence of stock market data, we have specifically focused on the development of multivariate time series regression techniques.

Therefore, the aim of this paper is:

- To evaluate the impact of technical indicators on the quality of S&P 500 price forecasts using multivariate time series regression models such as XGboost, LSTM, Random Forest, and Support Vector regression.
- To identify the most important technical indicators for a given forecasting model.
- To find the most important technical indicators for each category of volume, volatility, trend, and momentum.

The paper is organized as follows: Section 2 provides a wide range of literature reviews. In Section 3 data and methods implemented for this research are described. Section 4 explains the experimental results and findings of the research. Finally, Section 5 provides the research conclusions.

2. Literature review

The literature review examines the significance of technical indicators in predicting stock prices. However, we have identified a notable gap in the detection, highlighting, and categorization of the most crucial indicators for forecasting stock market prices, specifically for the S&P 500 stock index. Additionally, prior research has primarily focused on classification machine learning models, with limited studies utilizing regression machine learning techniques.

The academic debate on stock price predictability has evolved significantly with the integration of machine learning (ML). Early studies demonstrated that technical indicators when combined with ML models, could challenge the efficient market hypothesis (Patel et al., 2015). For instance, Patel et al. (2015) compared Artificial Neural Networks (ANN), Support Vector Machines (SVM), Random Forest (RF), and Naïve Bayes using 10 technical parameters on Indian indices (2004–2012), finding RF superior in handling continuous technical indicator values. Similarly, Kumar et al. (2016) introduced hybrid models integrating feature selection (e.g., Linear Correlation, Random Forest) with Proximal SVM (PSVM) on 12 global indices (2008–2013), identifying RF-PSVM as optimal. These works established ML's potential to decode market patterns using technical indicators, though they focused on classification tasks rather than precise price prediction.

To enhance predictive accuracy, researchers began merging ML with evolutionary algorithms. Qiu and Song (2016) optimized ANN weights using Genetic Algorithms (GA) to forecast the Nikkei 225 (1993–2013), achieving an 81.27% directional accuracy with 9 technical indicators. Chen and Hao (2017) further advanced this by combining feature-weighted SVM and KNN on Chinese indices (2008–2014), where Information Gain-selected technical indicators improved model robustness. Chung and Shin (2018) pioneered GA-LSTM hybrids for the Korean Stock Price Index (2000–2016), using GA to optimize LSTM hyperparameters and demonstrating superior performance over benchmarks. These studies highlighted the synergy between evolutionary algorithms and ML in refining technical indicator utility.

The advent of deep learning revolutionized feature engineering, enabling models to process vast sets of technical indicators. Song et al. (2019) constructed a Deep Neural Network (DNN) with 715 technical-derived features on Korean stock data (1990–2016), achieving state-of-the-art accuracy through custom filtering techniques. Naik and Mohan (2019) applied Boruta feature selection to 33 technical indicators for classifying Indian stock movements (2008–2018), with deep learning outperforming traditional ML. Nabipour et al. (2020) further validated deep learning's edge, showing RNN and LSTM dominance over 10 ML models on Iranian stocks (2009–2019) using 10 technical indicators. These efforts underscored deep learning's capacity to harness high-dimensional technical data.

As technical indicator sets grew, feature selection became critical to avoid overfitting. Yuan et al. (2020) employed Recursive Feature Elimination (RFE) on 60 features (including 4 technical indicators) for Chinese A-shares (2011–2018), identifying RF as the most robust model. Ampomah et al. (2021) combined PCA with tree-based ensembles on 40 indicators for NYSE/NASDAQ stocks (2005–2021), while Sakhare et al. (2023) prioritized 75 Blockchain-derived technical parameters via ensemble ranking for NIFTY 50 (1999–2019). These studies emphasized that curated subsets of technical indicators—not sheer volume—drive model efficacy.

Despite progress, few studies systematically categorized technical indicators. Alsubaie et al. (2019) tested 50 indicators on 99 stocks (2015–2018), noting performance declines beyond 30 features but lacking thematic grouping. Yun et al. (2021) expanded features to 67 for the Korean index (1993–2017) but treated them as homogenous inputs. Ji et al. (2022) improved 18 wavelet-denoised indicators for global indices (2005–2021) but did not classify them by financial function (e.g., momentum, volatility). This oversight limited interpretability and practical application for investors.

Most literature prioritizes classification (e.g., predicting price direction). For example, Basak et al. (2019) used XGBoost on 6 technical indicators to classify US/Indian stock movements (up to 2017), achieving high accuracy but neglecting continuous price prediction. Similarly, Ku et al. (2023) fused investor domain knowledge with 22 technical indicators in LSTM for Malaysian stocks (up to 2022), focusing on directional outcomes. Regression-based approaches, critical for quantitative trading strategies, remain underexplored (Fernández et al., 2023).

Existing studies disproportionately target emerging or niche markets (e.g., Korean, Iranian, Indian indices), with limited focus on the S&P 500—a global equity benchmark. Fernández et al. (2023) applied ARIMA to the S&P 500 (1960–2018) but used only 14 technical indicators, while Zhong and Hitchcock (2021) combined RF-LSTM on S&P 500 data (2000–2019) without categorizing indicators. This gap leaves the S&P 500's unique dynamics understudied in the context of technical indicator importance.

Collectively, prior work establishes ML's utility in leveraging technical indicators but reveals three critical gaps:

- Overemphasis on classification: Regression-based price forecasting is rare.
- Lack of indicator categorization: Technical indicators are rarely grouped into interpretable classes (e.g., momentum, trend).
- S&P 500 specificity: The index's predictive dynamics remain underexplored.

This study bridges these gaps by analyzing 88 technical indicators categorized into momentum, trend, volatility, and volume classes for regression-based S&P 500 forecasting. Unlike prior classification-centric work, we employ XGBoost, Random Forest, Support Vector Regression, and LSTM to predict daily prices (with/without lagged close data), augmented by PCA for dimensionality reduction. Our results identify the most impactful indicators per category, offering a structured framework for investors and advancing ML-driven financial analytics.

Overall, this research showcases the potential of integrating investor domain knowledge with advanced machine learning techniques like LSTM to make more accurate predictions in the stock market. A summary of reviewed studies based on the number of technical indicators, best-performing models, datasets, and the timeframe for analyzing their data is described in Table 1.

Table 1
Analysis of reviewed studies based on the number of indicators, best-performing models, dataset, and timeframe.

Reference	Technical Indicators	Best Performing Model	Dataset	Timeframe
P. S. Thakkar and K. Kotecha 2015	10	Random Forest	CNX Nifty, S&P BSE Sensex, Infosys Ltd. Reliance Industries	2003-2012
Kumar et al. 2016	55	Random Forest-PSVM	12 stock market indices	2008-2013
Gorenc N. and Velušček 2016	13	SVM-Guided D-threshold	370 S&P 500 companies	2004-2013
Qiu and Song 2016	9	GA-ANN	Nikkei stock index	2007-2013
Chen and Hao 2017	9	FWSVM	Shanghai and Shenzhen stock indices	2008-2014
Chung and Shin 2018	10	GA-LSTM	Korean stock price index	2000-2016
Weng et al. 2018	8	Boosted Regression Tree	Citi Group stock	2013-2016
Basak et al. 2019	6	XGBoost	10 US & Indian stocks	Begin - 2017
Song et al. 2019	715	Deep Neural Networks	Korean stock price index	1990-2016
Naik and Mohan 2019	10	ANN	Indian stock index	2008-2018
Alsubaie et al. 2019	50	Cost-Sensitive Naïve Bayes	99 Stock market indices	2015-2018
Ntakaris et al. 2019	52	LSTM AE	5 Nordic & Amazon and Google	2010 & 2015
Yuan et al. 2020	4	RFE-Random Forest	Chinese A-share stocks	2012-2019
Nabipour et al. 2020	10	RNN & LSTM	4 Iranian stocks	2009-2019
Botunac et al. 2020	9	LSTM	Apple, Microsoft & Facebook	2015-2019
Zhong and Hitchcock 2021	7	Random Forest-LSTM	S&P500	2000-2019
Yun et al. 2021	67	GA-XGBoost	Korean stock price index	1993-2017
Ampomah et al. 2021	40	AdaBoost of Bagging	NYSE, NASDAQ and NSE	2005-2019
Peng et al. 2021	124	SFS-Deep Neural Networks	Seven global market indices	2008-2019
Ji et al. 2022	18	FS-Random Forest	SSEC, HIS, DJI & S&P 500 indices	2005-2021
Chandar 2022	10	TI – CNN	NASDAQ and NYSE	2009-2018
Sakhare et al. 2023	75	History Bits model	NIFTY 50	1999-2019
Ku et al. 2023	22	Domain Knowledge + LSTM	100 Bursa Malaysia stocks	Begin -2022
Fernández et al. 2023	14	ARIMA	S&P 500	1960-2018

3. Methodology

This section describes the methodology used to find key technical indicators for S&P 500 index prediction. Section 3.1 defines technical indicators and their formula and explains the role of each indicator in stock price technical analysis. Data preparation and preprocessing are discussed in Section 3.2. In Section 3.3, feature selection using Principal Component Analysis (PCA) is described. Random Forest Regression, XGBoost model, Support Vector Regression, and Long Short-Term Memory models are briefly explained in Section 3.4 as the machine learning (ML) techniques applied in this research. Finally, in Section 3.5, three evaluation metrics that are implemented in this paper are described. The summary of the research methodology used in this paper is described in Fig. 1.

This framework is designed to systematically identify key indicators that significantly contribute to the performance of various machine learning models used for forecasting stock prices, specifically for the S&P 500 index. By employing a series of preprocessing techniques, model training strategies, and performance evaluation methods, this framework aims to ensure that the selected indicators not only demonstrate predictive power but also maintain interpretability through feature ranking methodologies.

The framework follows a structured approach:

Data Preprocessing: Initialize by addressing data quality issues including missing values and outliers, while normalizing the data to

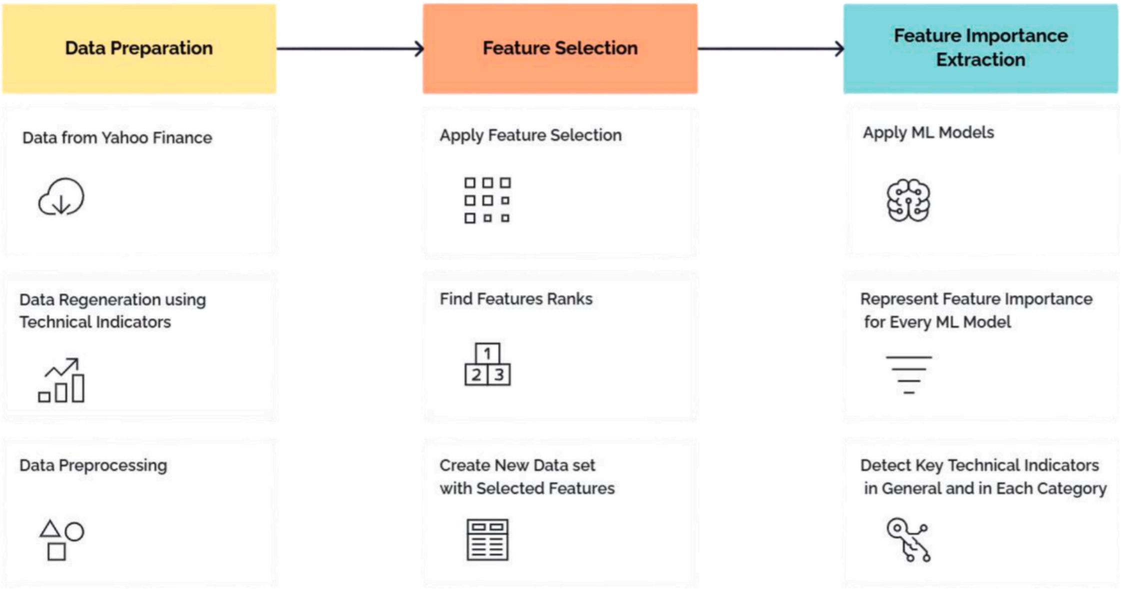
Data Preparation
Feature Selection
Feature Importance Extraction
Data from Yahoo Finance
Apply Feature Selection
Apply ML Models
Data Regeneration using Technical Indicators
Find Features Ranks
1
2
3
Represent Feature Importance for Every ML Model
Data Preprocessing
Create New Data set with Selected Features
Detect Key Technical Indicators in General and in Each Category

Fig. 1. Proposed machine learning (ML) framework for feature importance detection.

ensure it is suitable for model training.

Data Partitioning: Split the historical data into training, validation, and testing sets using a rolling window method to mimic real-world trading scenarios.

Model Training and Evaluation: Implement a variety of models, employing hyperparameter optimization and validation techniques to enhance predictive accuracy.

Feature Importance Analysis: Utilize SHAP values and permutation importance to determine the significance of each technical indicator across models.

Consensus Indicator Selection: Finally, aggregate the results to ascertain the most impactful indicators that consistently appear across multiple models.

The pseudocode below encapsulates the steps involved in this comprehensive framework, ensuring a methodical approach to technical indicator selection.

```
Algorithm: Technical Indicator Selection Framework
Input: Historical OHLCV data for S&P 500
Output: Selected key indicators, model performance metrics
1. Preprocess data:
    - Handle missing values via linear interpolation.
    - Remove outliers using IQR-based filtering.
    - Normalize features via Min-Max scaling.
    - Apply PCA (retain 95% variance) for dimensionality reduction.
2. Split data into training (70%), validation (15%), and testing (15%) with rolling
   windows (window=60 days, step=1).
3. For each model (SVR, XGBoost, RF, LSTM):
    a. Tune hyperparameters via Bayesian optimization (50 iterations).
    b. Train on lagged (Close(t-1)) and non-lagged (Close(t)) targets.
    c. Validate using walk-forward cross-validation.
4. Compute permutation importance for feature ranking.
5. 5. Aggregate results to identify consensus indicators (appearing in $\geq$ 3 models).
```

3.1. Experiment setup details

In our research, all the models are applied twice. the dataset spans 01/01/1950–12/07/2023, comprising 18,137 trading days. The target variable is the daily closing price of the S&P 500, both with and without a one-day lag. Preprocessing reduced the feature set from 88 to 35 indicators post-PCA. Following this step, the dataset is divided into two sets, a training set and a testing set. The training dataset contains trading days from January 1950 to June 2012 and the testing dataset contains trading days from June 2012 to June 2023. The relative size ratio between training and testing data sets is approximately 6:1.

To enhance the realism of the forecasting process, a rolling window approach was adopted with a 60-day lookback period and a 1-day step size. This method allows the model to train on the most recent 60 days of data, updating daily to reflect new information. Such a design mimics the dynamics of real-time trading, where decisions are made based on the latest available data, thus providing insights into the model's performance in a practical setting.

For the machine learning models employed, specific hyperparameters were optimized to enhance performance. The XGBoost model was configured with 200 estimators, a maximum depth of 6, and a learning rate of 0.01. These parameters were chosen to balance model complexity and training stability. The LSTM model was structured with two layers, each containing 64 units, and a dropout rate of 0.2 to mitigate overfitting. The model was trained for 100 epochs, allowing sufficient iterations for convergence while monitoring performance on the validation set.

3.2. Preprocessing phase expansion

The dataset exhibited a 2.1% incidence of missing values, particularly in the Average True Range (ATR) and Relative Vigor Index (RVI) indicators during the early 2010s. To address this issue, linear interpolation was applied, ensuring continuity in the dataset and preserving the integrity of the time series for subsequent analysis.

An outlier analysis revealed that approximately 1.5% of extreme values were present in the dataset, notably influenced by significant market events such as the 2015 Swiss Franc (CHF) crisis and the 2020 COVID-19 market crash. These outliers were removed using an Interquartile Range (IQR) method, which effectively mitigated their potential distortion of model training and evaluation.

To streamline the feature set and enhance model interpretability, 12 low-variance indicators, such as the Chaikin Oscillator, were excluded from the analysis. Additionally, 8 highly correlated features (with Pearson's correlation coefficient $|r| > 0.95$) were also removed to prevent multicollinearity, ensuring that the remaining features provided unique and valuable information to the models.

3.3. Algorithm selection justification

The selection of algorithms for this study was driven by their distinct advantages in addressing the complexities of financial time series forecasting (Zhang et al., 2024). Support Vector Regression (SVR) serves as a baseline model, effectively capturing linear relationships while offering kernel flexibility to handle non-linear patterns. Both XGBoost and Random Forest (RF) were chosen for their ensemble nature, providing robustness against noise and the ability to model non-linearities effectively (Zhao et al., 2024). The Long Short-Term Memory (LSTM) network was included for its capacity to capture temporal dependencies inherent in financial sequences, making it particularly suitable for time-series data (Nazareth & Reddy, 2023). This broad scope of algorithm selection ensures that the findings are model agnostic, thereby mitigating the risk of overreliance on a single technique (Pérez-Hernández & Arévalo-de-Pablos, 2024)

3.4. *Proposed approach limitations*

A notable limitation of this approach is the potential for temporal bias, as the models were trained on data spanning 2010 to 2023. This may limit their adaptability to structural market shifts, such as those experienced during high-inflation regimes, which could alter the underlying relationships in the data.

While Principal Component Analysis (PCA) was employed to retain 95% of the variance in the dataset, this approach carries the risk of losing granularity in low-variance indicators. Such indicators, although less prominent, may still provide critical insights that could enhance model performance.

The complexity of the LSTM model presents another challenge, as it requires extensive hyperparameter tuning to achieve optimal performance. Without appropriate regularization techniques, such as dropout, there is a heightened risk of overfitting, which could undermine the model's generalizability to unseen data.

3.5. *Technical indicators*

In essence, technical analysis is rooted in the belief that stock prices follow trends. This belief is based on the idea that investors collectively engage in patterned trading behavior, which ultimately determines stock price fluctuations (Fang, Qin, and Jacobsen, 2014). Unlike other factors like profit, margin, revenue, or earnings, these trends are statistical estimates derived from the price or value of a share. Technical analysts rely on historical data to identify price patterns, which encompass trends, momentum, volatility, and volume (Ahn et al., 2003; Bâra & Oprea, 2024; Jiang et al., 2020; Salim & Djunaidy, 2024). Active stock traders often employ technical indicators to analyze short-term price movements, while long-term investors use them to identify opportune moments to buy or sell. By combining technical analysis with trading systems, it become a valuable tool for forecasting future stock prices (Kumbure et al., 2022).

Technical indicators can be divided into 4 unique groups, with individual philosophies on how prices can be forecasted. These 4 groups are trending or moving averages, volatility, momentum, and volume indicators. Every one of these groups has in common the goal of uncovering insight into the future prices of the assets (van der Hagen, 2021).

The technical indicators generated in this paper in the data preparation section (Section 3.1) are listed in Appendix A. The 88 technical indicators used as input variables for feature selection in Section 3.3 are generated using Open, Close, High, and Low prices and Volume information from Yahoo Finance. The Python library used to generate technical indicators is TA-Lib. All indicators were subjected to Principal Component Analysis (PCA) for dimensionality reduction. Following feature selection, 35 indicators were retained and are described in the subsequent table. In addition, the formula of the most important technical indicators selected in the feature selection section is described as well. Note that some of the indicators have multiple outputs which are highlighted in bold in the table, (Table 2).

3.6. *Data normalization*

To apply the machine learning (ML) model to the technical indicators, data pre-processing for the input data is necessary. Data pre-processing employs data cleaning and normalization in this study. For the data cleaning step, unwanted and unnecessary features are removed from the feature set. The adjusted closing price is removed as it is redundant, and the closing prices contain the required information.

Following the data cleaning step, the data normalization step is required for the prediction. Normalization of the data makes it more amenable to the task at hand, for the prediction as described in Jin et al. (2023). Furthermore, data normalization rescales the technical indicators and prices to be used efficiently in the prediction model. Specifically, the normalization of input feature values helps the gradient descent converge much faster (Chung & Shin 2018) and preserves precisely all relationships in the data, and thereby, it does not introduce any bias. In this paper we have applied data normalization using the min-max formula in Eq. (1):

$$x' = (x - x_{min})/(x_{max} - x_{min}) \tag{1}$$

3.7. *Feature selection with Principal Component Analysis (PCA)*

Principal Component Analysis (PCA) was applied to reduce the dimensionality of the 88 technical indicators while retaining 95% of the dataset variance. This retained 35 principal components, balancing interpretability and computational efficiency. The PCA implementation addressed challenges such as high correlation (Pearson's $|r| > 0.95$) among features and missing values (2.1% in early 2010s data), resolved via linear interpolation. Standardization was performed using $Z_{ij} = (x_{ij} - \overline{x}_j)/s_j$, and the correlation matrix $R_j^2 = Z^{\mathrm{T}}Z/(n-1)$ was diagonalized to derive uncorrelated components. While PCA mitigated multicollinearity, it risked losing granularity in low-variance indicators like the Chaikin Oscillator, which were excluded post-analysis. For foundational PCA theory, see Htun et al. (2023) (Ji et al. 2022).

3.8. *Machine learning (ML) techniques*

Based on the literature, four machine learning (ML) techniques are selected for the implementation of the S&P 500 stock index. The description of each model is explained in the following sections.

3.8.1. *Random forest regression*

Random forest is an ensemble learning model that constructs multiple decision trees using bagging. It treats a collection, or "forest," of independent regression trees $\{t(k) : k = 1. 2.....B\}$ as base learners, where each tree is built from bootstrap samples drawn with replacements from the training dataset (Yin et al., 2023).

This nonparametric regression technique evaluates each tree's prediction error based on metrics like mean squared error (MSE), calculated using Out-Of-Bag (OOB) samples (Park, Kim, & Kim, 2022). The overall prediction at a new site, $\widehat{f}_{\varphi}(\theta)$, with predictor vector x, is determined by estimating the mean of all regression trees, as shown in the equation:

$$\widehat{f}_{\varphi}(x) = 1 \Big/ B \sum_{k=1}^{B} \widehat{f}^{(k)}(x;\ \theta_k) \tag{2}$$

where the variable θ_k determines which predictors are included in the k$^{\text{th}}$ tree with the aim of reducing variance and improving predictive performance compared to a single regression tree (Prasad & Bakhshi 2022).

In this research, the Random Forest model incorporated 500 trees with $m = \sqrt{p}$ variables per split to minimize Out-Of-Bag (OOB) error, effectively enhancing accuracy and robustness through the aggregation of predictions. This approach allows the model to reduce variance and

Table 2
Technical indicators formula with descriptions.

Indicator	Descriptions	Formula
SMA	The simple moving average (SMA) is a technical indicator used to determine whether the price of an asset will continue or reverse its uptrend or downtrend (Bâra & Oprea, 2024; Salim & Djunaidy, 2024).	$\mathbf{SMA_t(n)} = 1/n \sum_{i=1}^{n} \mathrm{Close}_{t-i}$
EMA	The exponential moving average (EMA) places more emphasis on the latest data points, allowing for quicker responses to current information (Bâra & Oprea, 2024; Salim & Djunaidy, 2024).	$\mathbf{EMA}_t(\mathbf{n}) = (\mathrm{Close}_t - \mathrm{EMA}_{t-1}(\mathrm{n})) * (\mathrm{SF}) + \mathrm{EMA}_{t-1}(n)$ $\mathrm{SF} = \text{Smoothin Factor}: (2 / (\mathrm{n} + 1))$ $\mathrm{EMA}_{t-1}(\mathrm{n}) = \text{EMA of previous day}$
TEMA	The Triple Exponential Moving Average (TEMA) is a trend-following indicator designed to minimize lag. TEMA offers a more responsive representation of price direction by applying multiple exponential smoothing calculations and offsetting the resulting delay (Salim & Djunaidy, 2024).	$\mathbf{TEMA}_t = 3\mathrm{EMA}_{t-1} - 3\mathrm{EMA}_{t-2} + \mathrm{EMA}_{t-3}$
WMA	A Weighted Moving Average (WMA) is a technical analysis tool that places greater emphasis on recent price data, thereby providing a more responsive indication of prevailing market trends compared to a SMA (Salim & Djunaidy, 2024).	$\mathbf{WMA}_t(\mathbf{Close}, \boldsymbol{n}) = (n \times \mathrm{Close}_t + (\mathrm{n} - 1) \times \mathrm{Close}_{t-1} + \ldots + \mathrm{Close}_{t-n})/[n \times (n + 1)/2]$
HMA	The Hull Moving Average (HMA) is a sophisticated technical indicator designed to swiftly identify emerging trends and potential trading opportunities (Salim & Djunaidy, 2024).	$\mathbf{HMA}_t = WMA\left(\left(2\mathrm{WMA}_t\left(\frac{n}{2}\right) - \mathrm{WMA}_t(n)\right), \sqrt{n}\right)$
HWMA	The Holt-Winters Moving Average (HWMA) is a technique that employs a three-parameter model. These parameters must be carefully chosen to accurately predict future values (Salim & Djunaidy, 2024).	$\mathbf{HWMA}_t = \mathrm{F}_t + \mathrm{V}_t + 0.5 * \mathrm{A}_t$ $\mathrm{F}_t = (1 - \mathrm{n_A}) * (\mathrm{F}_{t-1} + \mathrm{V}_{t-1} + 0.5 * \mathrm{A}_{t-1}) + \mathrm{n_A} * \mathrm{Close}_t$ $\mathrm{V}_t = (1 - \mathrm{n_B}) * (\mathrm{V}_{t-1} + \mathrm{A}_{t-1}) + \mathrm{n_B} * (\mathrm{F}_t - \mathrm{F}_{t-1})$ $\mathrm{A}_t = (1 - \mathrm{n_C}) * \mathrm{A}_{t-1} + \mathrm{n_C} * (\mathrm{V}_t - \mathrm{V}_{t-1})$ n_A: parameter that describes a smoothed series [0, 1] $\mathrm{n_B}$: parameter to assess the trend [0, 1] $\mathrm{n_C}$: parameter to assess seasonality [0, 1]
FWMA	The Fibonacci Weighted Moving Average (FWMA) is a technical indicator that uniquely combines the smoothing properties of traditional moving averages with the weighting principles derived from the Fibonacci sequence.	$\mathbf{FWMA}_t = (\mathrm{Close}_{t-1} * \mathrm{F1} + \mathrm{Close}_{t-2} * \mathrm{F2} + \ldots + \mathrm{Close}_{t-n} * \mathrm{Fn}) / (\mathrm{F1} + \mathrm{F2} + \ldots + \mathrm{Fn})$ F1, F2, ..., Fn are the corresponding Fibonacci numbers
SWMA	The Symmetric Weighted Moving Average (SWMA) provides adaptable insights into market trends through its dynamic calculation and variable length.	SWMA with the length of 4 bars: $\mathbf{SWMA} = (\mathrm{Close}_t + 2\mathrm{Close}_{t-1} + 2\mathrm{Close}_{t-2} + \mathrm{Close}_{t-3})/6$
HLC3	HLC3 is a simple technical indicator that calculates the average of the High, Low, and Close prices of a specific period. It's often used as a smoothing technique to reduce price volatility.	$\mathbf{HLC3}_t = (\mathrm{High}_t + \mathrm{Low}_t + \mathrm{Close}_t) / 3$
ATR	The Average True Range (ATR) measures price volatility, larger true ranges usually indicate strong back-and-forth movements. (Bâra & Oprea, 2024)	$\mathbf{ATR}_t(\mathrm{n}) = [(\mathrm{ATR}_{t-1} * (\mathrm{n} - 1) + \mathrm{TR}_t] / \mathrm{n}$ $\mathrm{TR}_t = \mathrm{Max}\ [(\mathrm{High}_t - \mathrm{Low}_t).\lvert \mathrm{High}_t - \mathrm{Close}_t \rvert.\lvert \mathrm{Low}_t - \mathrm{Close}_t \rvert]$ $\mathbf{ATR}_0 = \frac{1}{n} \sum_{i}^{n} \mathrm{TR_i}$
UL	The Ulcer (UL) is a volatility indicator that assesses downside risk by measuring the size and duration of price drops. Typically, this indicator is considered over 14 days. The Ulcer Index reveals the expected percentage decline from the highest price during this time frame.	$\textbf{Ulcer Index} = \sqrt{\text{Squared Average}}$ $\text{Squared Average} = \mathrm{SPD}_{14}^2/14$ $\mathrm{SPD}_{14} = \text{14 period Sum of PercentDrawdown}$ $\text{PercentDrawdown} = ((\mathrm{Close}_t - \mathrm{Close}_{14}^{\mathrm{m}}) / \mathrm{Close}_{14}^{\mathrm{m}}) * 100$ $\mathrm{Close}_{14}^{\mathrm{m}} = \text{14 period Max Close}$
BB	Bollinger Bands (BB) are volatility bands placed above and below a moving average. They are based on standard deviation and adjust as volatility levels change. These bands consist of a central band, as well as an upper and lower band. When the price reaches the upper band, it may indicate an overbought signal, while reaching the lower band may suggest an oversold signal. (Bâra & Oprea, 2024)	$\mathbf{UB}_t(\mathbf{n}) = \text{Upper Band} = \mathrm{SMA}_t(\mathrm{n}) + (2 * \mathrm{std}_t(\mathrm{n}))$ $\mathbf{LB}_t(\mathbf{n}) = \text{Lower Band} = \mathrm{SMA}_t(\mathrm{n}) - (2 * \mathrm{std}_t(\mathrm{n}))$ $\textbf{Middle Band}_t(\mathbf{n}) = \mathrm{SMA}_t(\mathrm{n})$ $\mathrm{std}_t(\mathrm{n}) = \text{standard deviation of n days } \mathrm{Close}_t$
DC	Donchian Channels establish the correlation between current prices and trading ranges during specific periods. The upper and lower lines signify the areas of bullish and bearish sentiment and highlight the highest and lowest price points reached during the corresponding periods of conflict.	$\mathbf{UC}_t(\mathbf{n}) = \text{Highest High in last n periods}$ $\mathbf{LC}_t(\mathbf{n}) = \text{Lowest Low in last n periods}$ $\textbf{Middle Channel}_t(\mathbf{n}) = ((\mathrm{UC}_t(\mathrm{n}) + \mathrm{LC}_t(\mathrm{n}))/2)$ $\text{Period} = \text{Minutes. hours. days. weeks. months}$
KC	Keltner Channels serve as volatility-based boundaries above and below the exponential moving average. These channels incorporate the Average True Range (ATR) rather than the standard deviation to determine the distance of the channels.	$\mathbf{ML} = \text{Middle Line} = \text{Exponential moving average}$ $\mathbf{UL} = \text{Upper Line} = \mathrm{EMA}_t(\mathrm{n}) + (2 * \mathrm{ATR}_t(\mathrm{n}))$ $\mathbf{LL} = \text{Lower Line} = \mathrm{EMA}_t(\mathrm{n}) - (2 * \mathrm{ATR}_t(\mathrm{n}))$
ROC	The Rate of Change indicator (ROC), commonly referred to as "Momentum", measures the percentage change in price over a designated time frame. Its purpose is to identify signals of overbought and oversold conditions in the market.	$\mathbf{ROC}_t(\mathbf{n}) = [(\mathrm{Close}_t - \mathrm{Close}_{t-n}) / (\mathrm{Close}_{t-n})] * 100$ $\mathrm{Close}_{t-n} = \text{Close of past n periods}$
KST	Know Sure Thing (KST) is a momentum indicator used to interpret price data. It produces a trading signal when it crosses the signal line. Additionally, investors seek out overbought or oversold conditions, often combining the KST with other technical indicators to maximize their profits.	$\textbf{KST Signal Line} = \mathrm{SMA}_t(9) \text{ of KST}$ $\mathrm{KST}_t = (\mathrm{R1}_t \times 1) + (\mathrm{R2}_t \times 2) + (\mathrm{R3}_t \times 3) + (\mathrm{R4}_t \times 4)$ $\mathrm{R1}_t = \mathrm{SMA}_t(10) \text{ of } \mathrm{ROC}_t(10)$ $\mathrm{R2}_t = \mathrm{SMA}_t(10) \text{ of } \mathrm{ROC}_t(15)$ $\mathrm{R3}_t = \mathrm{SMA}_t(10) \text{ of } \mathrm{ROC}_t(20)$ $\mathrm{R4}_t = \mathrm{SMA}_t(15) \text{ of } \mathrm{ROC}_t(30)$
Ichimoku Cloud	The Ichimoku cloud, composed of five lines. Two of these lines create a shielded cloud, representing the difference between them. The position of the price relative to the cloud determines the trend: below the cloud indicates a bearish trend, while above the cloud signifies an upward trend.	**Tenkan sen** : (9 period high + 9 period low)/2)) **Kijun Sen** : (26 period high + 26 period low)/2)) **Senkou Span A** : (Tenkan Sen + Kijun Sen)/2)) **Senkou Span B** : (52 period high + 52 period low)/2)) **Chikou Span** : Close of the past 26 days
MACD	The moving average convergence divergence (MACD) is derived by subtracting the 26-period exponential moving average from the 12-period EMA. Additionally, the signal line is generated as the nine-period EMA (Bâra & Oprea, 2024).	**MACD Line** : $(\mathrm{EMA}_t(12) - \mathrm{EMA}_t(26))$ Signal Line : $\mathrm{EMA}_t(9)$ of MACD Line **MACD Histogram** : MACD Line − Signal Line
MI	The mass index utilizes high and low ranges to identify potential trend reversals based on the range's magnitude. If the range surpasses a certain threshold and then contracts, it suggests a probable reversal of the current trend. This makes the mass index an effective tool for short-term trading.	$\textbf{Mass Index} = \sum_{t=1}^{25} \mathrm{EMA}_t\mathrm{Ratio}$ $\mathrm{EMA}_t\mathrm{Ratio} = \text{Single } \mathrm{EMA}_t / \text{ Double } \mathrm{EMA}_t$ $\text{Single } \mathrm{EMA}_t = \mathrm{EMA}_t(9) \text{ of the HL.d}$ $\text{Double } \mathrm{EMA}_t = \mathrm{EMA}_t(9) \text{ of the } \mathrm{EMA}_t(9) \text{ of the HL.d}$ $\mathrm{HL.d} = \mathrm{High}_t - \mathrm{Low}_t$

(*continued on next page*)

Table 2 (*continued*)

Indicator	Descriptions	Formula
CCI	The Commodity Channel Index (CCI) is a measurement used to assess trend direction and strength in stocks. It compares the stock's price to its moving averages over a specific period of time. Additionally, it helps determine if a stock is nearing levels of being overbought or oversold.	$\mathbf{CCI}_t(\mathbf{n}) = (\mathrm{TP}_t(\mathrm{n}) - \mathrm{SMA}_t(\mathrm{n}) \text{ of } \mathrm{TP}_t(\mathrm{n})) \ / \ (0.015 * \mathrm{MD})$ $\mathrm{TP}_t(\mathrm{n}) = \sum_{i=1}^{n} (\mathrm{High}_{t-i} + \mathrm{Low}_{t-i} + \mathrm{Close}_{t-i})/3$ $\mathrm{MD}_t(\mathrm{n}) = \sum_{i=1}^{n} \lvert \mathrm{TP}_{t-i}(n) - \mathrm{SMA}_{t-i}(\mathrm{n}) \text{ of } \mathrm{TP}_{t-i}(n) \rvert / n$
PSAR	Traders rely on the Parabolic Stop and Reverse (PSAR) indicator to determine trend direction and the possibility of price reversals. By incorporating the last price extreme (EP) and an acceleration factor (AF), the PSAR indicator identifies where it will appear on a chart (Jiang et al., 2020)	$\textbf{Uptrend}: \text{ Prior PSAR} + \text{Prior AF (Prior EP} - - \text{ Prior PSAR)}$ $\textbf{Downtrend}: \text{ Prior PSAR} - - \text{ Prior AF (Prior PSAR} - - \text{ Prior EP)}$ $\mathrm{EP} = \text{HH for an uptrend and LL for a downtrend}$ $\mathrm{HH} = \text{Highest High. LL} = \text{Lowest Low}$ $\mathrm{AF} = \text{Default of 0.02 increasing by 0.02}$
Aroon	Two lines known as "Aroon Up" and "Aroon Down" play a role in understanding market dynamics. These lines range from zero to 100 and provide insights into the strength of both uptrends and downtrends. When the "Aroon Up" line surpasses or remains below the "Aroon Down" line, it indicates a bullish or bearish price action, respectively.	$\text{Aroon Up} = \left(\frac{\text{(number of days since n days High)}}{n}\right) * 100$ $\text{Aroon Down} = \left(\frac{\text{(number of days since n days Low)}}{n}\right) * 100$ $\textbf{Aroon Indicator} = \text{Aroon Up} - \text{Aroon Down}$
AO	The Awesome Oscillator (AO) is a tool used to measure market momentum by calculating the difference between 34 and 5 simple moving averages. Unlike traditional moving averages that are based on closing prices, these averages use the average candle price for each bar.	$\mathrm{AO}_t = \mathrm{SMA}_t(5) \text{ of } \mathrm{MP}_t - \mathrm{SMA}_t(34) \text{ of } \mathrm{MP}_t$ $\text{Median Price} = \mathrm{MP}_t = (\mathrm{High}_t + \mathrm{Low}_t)/2$
%R	Another popular momentum indicator is %R, which represents the reverse of the fast stochastic oscillator. %R reflects the Highest High of the look-back period.	$\%\mathbf{R}(\mathbf{n}) = (\mathrm{HH}_t(\mathrm{n}) - \mathrm{Close}_t) \ / \ (\mathrm{HH}_t(\mathrm{n}) - \mathrm{LL}_t(\mathrm{n})) * (-100)$ $\mathrm{HH}_t(\mathrm{n}) = \text{Highest High of n period}$ $\mathrm{HH}_t(\mathrm{n}) = \text{Lowest Low of n period}$
Price Distance	Assessment of the magnitude of price changes between the previous day's close and the current day's trading (Bâra & Oprea, 2024).	$\mathbf{PDIST}_t(\mathbf{n}) = 2(\mathrm{high}_t - \mathrm{low}_t) - \lvert \mathrm{close}_t - \mathrm{open}_t \rvert + \lvert \mathrm{open}_t - \mathrm{close}_{t-n} \rvert$
KAMA	For traders who want to account for market noise and volatility, Kaufman's adaptive moving average (KAMA) is a reliable option. KAMA actively monitors prices and adjusts accordingly when there are small price fluctuations and low noise levels. However, as price volatility increases, KAMA starts tracking prices from a greater distance (Salim & Djunaidy, 2024).	$\mathbf{KAMA}_t = \mathrm{KAMA}_{t-1} + \mathrm{SC} * (\mathrm{Price} - \mathrm{KAMA}_{t-1})$ $\mathrm{SC} = [\mathrm{ER} * (\text{Fastest SC} - \text{Slowest SC}) + \text{Slowest SC}]^2$ $\text{Fastest SC} = \text{The smoothing constant of fastest } \mathrm{EMA}_t(2)$ $\text{Slowest SC} = \text{The smoothing constant of slowest } \mathrm{EMA}_t(30)$
PPO	The Percentage Price Oscillator (PPO) is a momentum index that calculates the percentage difference between two moving averages, with the higher moving average as the reference point.	$\mathbf{PPO}_t : \left\{\frac{\mathrm{EMA}_t(12) - \mathrm{EMA}_t(26)}{\mathrm{EMA}_t(26)}\right\} * 100 \cdots \text{12 \& 26 are defult}$ $\text{Signal Line}: \mathrm{EMA}_t(9) \text{ of PPO}$ $\text{PPO Histogram}: \text{PPO} - \text{Signal Line}$
PVO	The Percentage Volume Oscillator (PVO) is a momentum oscillator specifically designed for volume. It measures the percentage difference between two volume-based moving averages, again using the higher moving average as the benchmark.	$\mathbf{PVO}_t : \left(\frac{\mathrm{VEMA}_t(12) - \mathrm{VEMA}_t(26)}{\mathrm{VEMA}_t(26)}\right) * 100$ $\mathrm{VEMA}_t(\mathrm{n}) = \mathrm{EMA}_t(\mathrm{n}) \text{of Volume}$ $\text{Signal Line}: \mathrm{EMA}_t(9) \text{ of PVO}$ $\text{PVO Histogram}: \text{PVO} - \text{Signal Line}$
PVT	PVT is a momentum-based indicator that measures the buying and selling pressure in the market by combining price and volume data. It helps identify potential trend reversals and confirms existing trends.	$\mathbf{PVT}_t = \mathrm{PVT}_{t-1} + (\mathrm{Close}_t - \mathrm{Close}_{t-1}) * \mathrm{Volume}_t$
RSI	The Relative Strength Index (RSI) serves as a momentum indicator that assesses the ratio of relative gain to relative loss. It determines the strength of price movements by comparing the number of positive and negative price changes (Salim & Djunaidy, 2024).	$\mathbf{RSI} = 100 - 100/(1 + \mathrm{RS})$ $\mathrm{RS} = \text{Average Gain} \ / \ \text{Average Loss}$ $\text{Average Gain} = (\mathrm{PAG} * (\mathrm{n} - 1) + \text{Current Gain}) / \mathrm{n}$ $\text{Average Loss} = (\mathrm{PAL} * (\mathrm{n} - 1) + \text{Current Loss}) \ / \ \mathrm{n}$ $\mathrm{PAL} = \text{Average Loss in past n-1 period}$ $\mathrm{PAG} = \text{Average Gain in past n-1 period}$
Stochastic	The stochastic oscillator is a tool used to determine the position of a stock's closing price relative to its high and low points over a certain period of time, usually around 14 days.	$\%\mathbf{K}(\mathbf{n}) = (\mathrm{Close}_t - \mathrm{LL}_t(\mathrm{n}))/(\mathrm{HH}_t(\mathrm{n}) - \mathrm{LL}_t(\mathrm{n})) * 100$ $\%\mathbf{D}(\mathbf{n}) = \mathrm{SMA}_t(3) \text{ of } \%\mathrm{K}(\mathrm{n})$ $\mathrm{LL}_t(\mathrm{n}) = \text{Lowest Low for the last n period}$ $\mathrm{HH}_t(\mathrm{n}) = \text{Highest High for the last n period}$
TSI	The True Strength Index (TSI) serves multiple purposes such as identifying levels of oversold and overbought conditions, signaling trend reversals through the intersection of a signal line, and plotting the strength of a trend using divergence. The TSI achieves this by smoothing out the price action, providing a more stable line that filters out any disruptions or volatility.	$\mathrm{TSI} = 100 * (\mathrm{DSP} \ / \ \mathrm{DSAP})$ $\mathrm{PC} = \mathrm{Close}_t - \mathrm{Close}_{t-1}$ $\mathrm{DSP} = \mathrm{EMA}_t(13) \text{ of } \mathrm{EMA}_t(25) \text{ of PC}$ $\mathrm{DSAP} = \mathrm{EMA}_t(13) \text{ of } \mathrm{EMA}_t(25) \text{ of } \lvert \mathrm{PC} \rvert$
ADX	The Average Directional Index (ADX) is a metric that utilizes the smoothed differences between the Positive Directional Indicator (DI+) and the Negative Directional Indicator (DI-) to gauge the direction and strength of a trend over a specific period of time. When DI+ and DI- intersect, it suggests a change in the direction of the stock price. (Jiang et al., 2020; Salim & Djunaidy, 2024)	$\mathbf{ADX}_t(\mathbf{n}) = \mathrm{ADX}_{t-1} * (\mathrm{n} - 1) + \mathrm{DX}_t$ $\mathrm{DX}_t = 100 * \left(\left\lvert \frac{\mathrm{DI}_t^+ - \mathrm{DI}_t^-}{\mathrm{DI}_t^+ + \mathrm{DI}_t^-} \right\rvert\right)$ $\mathrm{DI}_t^+ = 100 * \left(\mathrm{SMA}_t(\mathrm{n}) \text{ of } \mathrm{DM}_t^+\right) / \mathrm{ATR}_t$ $\mathrm{DI}_t^- = 100 * \left(\mathrm{SMA}_t(\mathrm{n}) \text{ of } \mathrm{DM}_t^-\right) / \mathrm{ATR}_t$ $\mathrm{DM}_t^+ = \mathrm{High}_t - \mathrm{High}_{t-1}$ $\mathrm{DM}_t^- = \mathrm{Low}_{t-1} - \mathrm{Low}_t$ $\textbf{Initial ADX}(\mathbf{n}) = \text{n period average of } \mathrm{DX}_\mathrm{t}$
FI	The Force Index is an indicator that utilizes price and volume to assess the strength of a movement or detect potential turning points.	$\textbf{Force Index}_\mathbf{n} = \mathrm{EMA}_t(\mathrm{n}) \text{ of } \mathrm{F}_t$ $\mathrm{F}_\mathrm{t} = [\mathrm{Close}_t - \mathrm{Close}_{t-1}] * \mathrm{Volume}_t$
EMV	Ease of Movement (EMV) examines the relationship between an asset's price change and its associated volume. It helps to understand how volume impacts price alterations and is particularly valuable for evaluating the strength of a trend.	$\mathrm{EMV}_\mathrm{t}(1) = \left(\frac{\mathrm{H}_\mathrm{t} + \mathrm{L}_\mathrm{t}}{2} - \frac{\mathrm{H}_{\mathrm{t}-1} + \mathrm{L}_{\mathrm{t}-1}}{2}\right) / \left(\frac{\mathrm{Vol}}{10^8} / (\mathrm{H}_\mathrm{t} - \mathrm{L}_\mathrm{t})\right)$ $\mathrm{H}_\mathrm{t} = \text{Current High, } \mathrm{H}_{\mathrm{t}-1} = \text{Previous High}$ $\mathrm{L}_\mathrm{t} = \text{Current Low, } \mathrm{L}_{\mathrm{t}-1} = \text{Previous Low}$ $\mathbf{EMV}_t(\mathbf{n}) = \mathrm{SMA}_t(\mathrm{n}) \text{ of } \mathrm{EMV}_t(1)$
MFI	The Money Flow Index (MFI) evaluates buying and selling pressure by considering both price and volume. When the typical price increases (indicating buying pressure), the MFI is positive; when the typical price decreases (indicating selling pressure), it is negative (Jiang et al., 2020; Salim & Djunaidy, 2024)	$\text{Money Flow}_t = \text{Typical Price}_t * \text{Volume}_t$ $\text{Money Flow Ratio}_t(\mathrm{n}) = \frac{\sum_{i=1}^{n} \text{Positive Money Flow}_{t-i}}{\sum_{i=1}^{n} \text{Negative Money Flow}_{t-i}}$ $\mathbf{MFI}_t(\mathbf{n}) = 100 - \frac{100}{1 + \text{Money Flow Ratio}_t(\mathrm{n})}$

(*continued on next page*)

Table 2 (*continued*)

Indicator	Descriptions	Formula
VPT	The Volume Price Trend (VPT) indicator connects price and volume by factoring in the current cumulative volume along with a multiple of the percentage change in the stock price and the current volume as it fluctuates.	$\mathrm{VPT}_t = \mathrm{VPT}_{\mathrm{t-1}} + \mathrm{Volum}_t * \frac{\mathrm{Close}_{\mathrm{t}} - \mathrm{Close}_{\mathrm{t-1}}}{\mathrm{Close}_{\mathrm{t-1}}}$ $\mathrm{VPT}_{\mathrm{t-1}} = \text{Previous VPT}$ $\mathrm{Close}_{\mathrm{t-1}} = \text{Previous Close}$
VWAP	Volume-weighted average Price (VWAP) is a straightforward concept – it is the average price of a security, weighted by its trading volume. This metric specifically pertains to the current trading day (Jiang et al., 2020)	$\mathbf{VWAP}\ _t(\mathbf{n}) = \sum_{i=1}^{n} (\mathrm{Volume}_{t-i} * \mathrm{TP}_{t-i}) / \sum_{i=1}^{n} \mathrm{Volume}_{t-i}$ $\mathrm{TP}_t = (\mathrm{High}_t + \mathrm{Low}_t + \mathrm{Close}_t)/3$
CMF	Chaikin Money Flow (CMF), on the other hand, assesses the flow of money over a defined time period. The CMF indicator oscillates either above or below the zero line. Traders utilize the CMF's absolute level to gauge the balance between buying and selling pressure (Jiang et al., 2020)	$\mathbf{CMF}_t(\mathbf{n}) = \sum_{i=1}^{n} \mathrm{MFV}_{t-1} \ / \sum_{i=1}^{n} \mathrm{Volume}_{t-1}$ $\mathrm{MFV}_t = \mathrm{MFM}_t * \mathrm{Volume}_t$ $\mathrm{MFM}_t = \frac{(\mathrm{Close}_t - \mathrm{Low}_t) - (\mathrm{High}_t - \mathrm{Close}_t)]}{\mathrm{High}_t - \mathrm{Low}_t}$
OBV	Moving on, On Balance Volume (OBV) serves as a cumulative indicator, measuring the overall buying and selling pressure. It increases when volume rises on up days and subtracts when volume declines on down days. OBV captures both positive and negative volume flows. Traders often examine the divergence between OBV and price to predict potential price changes or to validate an existing trend (Salim & Djunaidy, 2024).	If $\mathrm{Close}_{\mathrm{t}} > \mathrm{Close}_{\mathrm{t-1}}$: $\mathbf{OBV}\ _t = \mathrm{OBV}\ _{t-1} + \mathrm{Volume}_t$ If $\mathrm{Close}_{\mathrm{t}} < \mathrm{Close}_{\mathrm{t-1}}$: $\mathbf{OBV}\ _t = \mathrm{OBV}\ _{t-1} - \mathrm{Volume}_t$ If $\mathrm{Close}_{\mathrm{t}} = \mathrm{Close}_{\mathrm{t-1}}$: $\mathbf{OBV}\ _t = \mathrm{OBV}\ _{t-1}$

capture complex interactions in financial time series data. The model's performance, evaluated using Mean Absolute Error (MAE), resulted in a value of 1.39 without lagged variables, indicating strong predictive accuracy. By minimizing OOB error, the Random Forest model provides an unbiased estimate of its performance, making it a reliable choice for financial forecasting and emphasizing the importance of model evaluation.

3.8.2. *XGBoost model*

The XGBoost model, a type of gradient boosting method, was utilized to predict stock trends. This model is highly scalable and outperforms existing tree-based algorithms in terms of learning speed and prediction accuracy due to its parallelization and decentralization capabilities (Han, Kim, & Enke, 2023). It consists of multiple Classification and Regression Trees (CART), a approach proposed by (Breiman, Friedman, and Olshen n.d.). CART shares similarities with ID3 but instead uses the Gini index as a criterion for selecting variables, rather than entropy. It performs binary splits and generates a decision tree by segmenting a subset of the dataset using all predictors to create two child nodes. The XGBoost model, incorporating multiple CART, is defined as follows:

$$\widehat{y_i} = \sum_{j=1}^{J} f_j\ (x_i).\ f_j \in F. F = \{f(x) = w_{\ q(x)}\}\left(q : R^m \rightarrow T.\ w \in R^T\right) \tag{3}$$

where R is the number of trees and F denotes all possible CART. f_j corresponds to each independent tree and the weight of each leaf. The final prediction was performed by summing the scores of each leaf. The objective function of the XGBoost model is expressed as follows:

$$Obj = \sum_i i\ l(y_i.\ \widehat{y_i}) + \sum_j \Omega\left(f_j\right)\ .\ \Omega(f) = \gamma K + 0.5\ \lambda \|w\|^2 \tag{4}$$

where l is a loss function that measures the difference between y_i and $\widehat{y_i}$. And $\Omega\left(f_j\right)$ is a regularization term that prevents overfitting by adjusting the complexity of the model. γ is the parameter of complexity for the regularization term and K is the number of leaves. $\|w\|^2$is the L_2 norm of weight regularization and λ is a constant coefficient.

In this research, we configured the XGBoost model with 200 estimators, a maximum depth of 6, and a learning rate of 0.01 to effectively address non-linearities in the data while minimizing the risk of overfitting. To further enhance the model's generalization, we included a regularization term set at $\lambda = 0.5$, which penalizes the complexity of the trees. This combination of parameters supports the development of a robust forecasting model that balances performance with the need to prevent overfitting the training data.

3.8.3. *Support Vector Regression (SVR)*

Support Vector Regression (SVR) defines the regression function $f(x)$ based on input training patterns and their corresponding desired outputs. The function is represented as $f(x) = \omega x^T + b$, where ω is the weight vector, b is the intercept, and x is the input vector. The optimization objective consists of minimizing the expression $1/\ 2\ \|\omega\|^2 + C\sum_{i=1}^{n} \left(\xi_i + \xi_i^*\right)$ subject to constraints that ensure the estimated outputs remain close to the target values while allowing for some flexibility through slack variables ξ and ξ^*.

In SVR, the regularization parameter C balances the trade-off between estimation accuracy and model complexity, while $\epsilon\epsilon$ defines a margin of tolerance for predictions. When extending to non-linear regression, SVR employs a mapping function that transforms inputs into a higher-dimensional feature space using a kernel function $K(x,x')$. In this study, a Radial Basis Function (RBF) kernel is used, expressed as $K(x.,\ x') = exp\left(-\ \|x - x'\|^2\ /\sigma^2\right)$. This approach allows SVR to effectively capture non-linear relationships within the data while retaining robustness in its predictions.

This research utilized a Radial Basis Function (RBF) kernel with a parameter set at $\sigma = 0.1$ which enhances the model's ability to capture non-linear relationships in the data by tuning the kernel's sensitivity to individual data points. The regularization parameter $C = 1.0$ was selected to strike a balance between model complexity and minimizing classification errors. To optimize the model's performance, hyperparameters were adjusted through grid search, a methodical technique that explores different combinations of parameter values. This approach is particularly effective for modeling and predicting the non-linear price dynamics often encountered in financial data, enabling the SVR to effectively adapt to complex patterns.

3.8.4. *Long Short-Term Memory (LSTM)*

LSTM models, a type of Recurrent Neural Networks (RNN), are effective in various applications such as financial time series forecasting (Chung and Shin 2018; Kuber, Yadav, and Yadav 2022). These models can gather data from previous stages and use it for future predictions. One important feature of LSTM is its incorporation of gates for memorizing earlier stages along with a memory line. The composition of LSTM nodes is depicted in Fig. 2.

Each LSTM node comprises a set of cells responsible for storing passed data streams. The upper line in each cell functions as a transport line, facilitating data flow from the past to the present nodes. The independence of these cells enables the model to selectively filter and add values from one cell to another. Ultimately, the sigmoidal neural network layer, which forms the gates, guides the cell toward an optimal value by either disallowing or allowing data to pass through. Each sigmoid layer has a binary value (0 or 1), indicating whether to block or allow data transmission. The objective is to effectively control the state

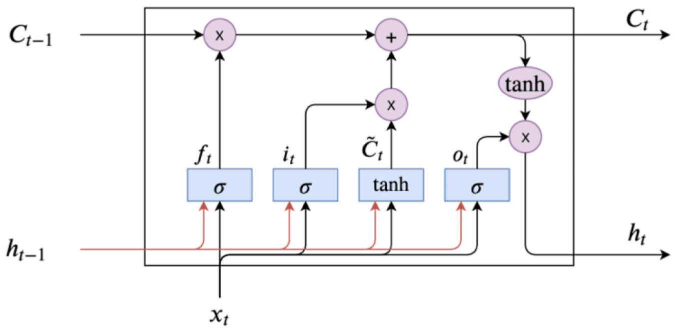

Fig. 2. The internal structure of the LSTM model (Smilevski, 2020).

of each cell. The gates are controlled according to the following procedure (Kuber et al., 2022):

Memory Gate (C_t) chooses which new data will be stored in the cell. First, a sigmoid layer "input door layer" chooses which values will be changed.

Input Gate (i_t):

$$i_t = \sigma(W_{ki}\cdot(X_t.\ h_{t-1}.\ C_{t-1}) + b_i) \tag{5}$$

Next, a *tanh* layer makes a vector of new candidate values that could be added to the state.

$$C_t = f_t\cdot C_{t-1} + i_t\cdot\tanh(W_{lc}(X_t.\ h_{t-1}) + b_c) \tag{6}$$

Forget Gate (f_t) outputs a number between 0 and 1, where 1 illustrates "completely keep this"; whereas 0 indicates "completely ignore this."

$$f_t = \sigma\big(W_{ef}\cdot(X_t.h_{t-1}) + W_{cf}C_{t-1} + b_f\big) \tag{7}$$

Output Gate (o_t) decides what will be the output of each cell. The output value will be based on the cell state along with the filtered and freshest added data.

$$o_t = \sigma(W_{ji}\cdot(X_t.\ h_{t-1}.\ C_{t-1}) + b_o \tag{8}$$

Where $W_{e*}\ .\ W_{c*}\ .\ W_{k*}\ .\ W_{l*}\ .\ W_{j*}$ and $b_f.\ b_i.\ b_c.\ b_o$ are the weight and bias of each layer respectively, and

$$h_t = o_t\cdot\tanh(C_t) \tag{9}$$

represents the hidden layer output.

In this research, we employed a two-layer Long Short-Term Memory (LSTM) architecture, consisting of 64 units per layer and incorporating a dropout rate of 0.2 to effectively capture the temporal dependencies in the S&P 500's daily prices. The model was trained for 100 epochs using a 60-day rolling window to simulate real-time trading conditions. To optimize the model's performance while balancing the risks of overfitting and convergence speed, we utilized Bayesian optimization over 50 iterations for parameter tuning. This approach aligns with established practices in the field, as noted by Goodfellow et al., 2016, ensuring robust model evaluation through the use of evaluation metrics.

3.9. Evaluation measures

Evaluation measures are crucial for assessing stock market forecasting model performance, enabling comparisons between model forecasts and actual values. For classification models, accuracy-based metrics are usually used, while regression models rely on error-based metrics like MAE and RMSE. In this research, we implemented regression models and compared their results using three established evaluation metrics: MAE, RMSE, and MAPE. Metrics were calculated on a 15% testing set (2012–2023) using walk-forward validation to reflect real-world trading conditions. Three metrics were selected for their relevance to financial forecasting (Obthong et al., 2020; Kumbure et al., 2022):

MAE (Mean Absolute Error): Prioritized for robustness to outliers.

$$\text{MAE} \ = 1\Big/n\sum|y_i - \widehat{y}_i|$$

RMSE (Root Mean Square **Error):** Emphasized larger errors, penalizing volatile mispredictions.

$$\text{RMSE} \ = \ \sqrt{\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2\Big/n}$$

MAPE (Mean Absolute Percentage Error): Provided relative error interpretation, though less reliable near zero.

$$\text{MAPE} \ = \ 1\Big/n\sum_{i=1}^{n}|y_i - \widehat{y}_i|\Big/y_i$$

4. Results and discussion

Fig. 3 presents the data used for the analysis. The original time series of the closing price of the S&P 500 index is depicted by the blue curve, with the vertical axis representing the closing price and the horizontal axis representing the interval of 01/01/1950–12/07/2023. The objective of this analysis is to identify key indicators that can accurately predict the future closing price of the S&P 500 index. Despite various irregularities, it is notable that the overall direction of the closing price is upwards.

In Section 3.2, we discussed how our dataset was collected and developed. We created technical indicators, removed null values, and normalized the results. Then, by Section 3.3, we applied the PCA feature selection method to eliminate redundant features from the dataset. This prepared the data for use in the machine learning techniques described in Section 3.4, namely Random Forest Regression, XGBoost, Support Vector Regression, and Long Short-Term Memory. This section presents the overall results in two parts. The first part, Section 4.1, provides a detailed analysis of the model evaluation using three evaluation metrics outlined in Section 3.5. The second part, Section 4.2, discusses the outcomes of the selected important features for each technique and category. It is worth noting that all experiments were conducted twice and then compared. The first experiment used the Close price of the current day as the dependent variable, while the second experiment utilized a one-day lag in the Close price.

4.1. Model evaluation

Model performance assessment aims to quantify the ability of the models developed in Section 3 on the testing dataset. Three statistical criteria (MAPE, MAE, and RMSE) are used to assess how well the resulting models performed for both datasets of Close price with and without lag. Table 3 represents the evaluation results for the different predictive models.

The comparative analysis of machine learning models reveals significant differences in predictive performance for the S&P 500 stock index based on three key metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE).

Support Vector Regression (SVR) models exhibited relatively high MAE values of 7.76 and 8.06, with corresponding MAPE scores of 2.04 and 2.08, indicating substantial inaccuracies in predictions. In contrast, Long Short-Term Memory (LSTM) models demonstrated remarkable accuracy, with the model without lag achieving an MAE of 0.014 and a MAPE of 0.008, while the lag-inclusive model produced slightly higher values of 0.051 for MAE and 0.085 for MAPE.

The Random Forest model outperformed SVR but fell short of LSTM, yielding MAE values of 1.39 (without lag) and 2.08 (with lag), along with MAPE scores of 0.017 and 0.010, respectively. XGBoost, however,

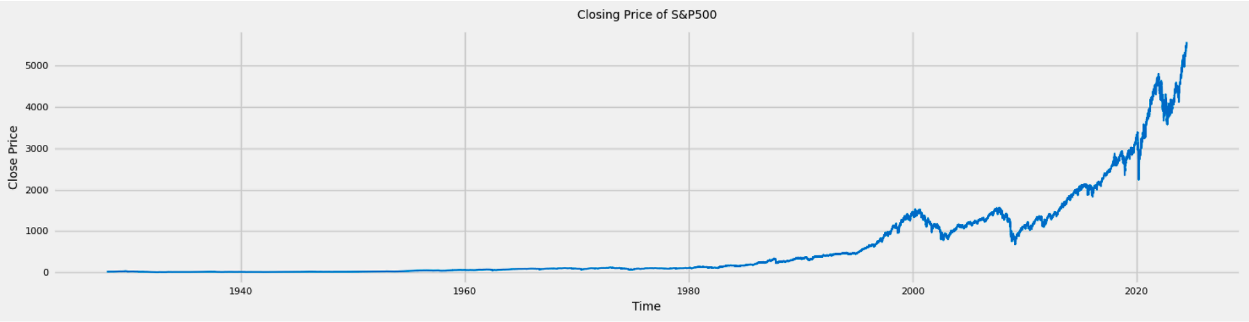

Fig. 3. S&P 500 closing price along with moving averages.

Table 3
Evaluation metrics of machine learning (ML) models for predicting S&P 500 stock index movements.

Machine Learning Models	MAE	MAPE	RMSE
Support Vector Regression (without lag in price)	7.76	2.04	10.48
Support Vector Regression (with lag in price)	8.06	2.08	10.92
LSTM (without lag in price)	0.014	0.008	0.045
LSTM (with lag in price)	0.051	0.085	0.19
Random Forest (without lag in price)	1.39	0.017	4.47
Random Forest (with lag in price)	2.08	0.010	5.21
XGBoost (without lag in price)	4.89	0.084	9.24
XGBoost (with lag in price)	18.92	0.894	21.44

performed the poorest, with MAE reaching 4.89 without lag and surging to 18.92 with lag, accompanied by MAPE values of 0.084 and 0.894. The RMSE metrics further support these findings, with LSTM models recording 0.045 (without lag) and 0.19 (with lag), Random Forest showing 4.47 and 5.21, while XGBoost's performance declined from 9.24 to 21.44 when lag was considered.

In conclusion, these results indicate that LSTM is the superior method for forecasting S&P500 index movements, significantly outperforming both SVR and XGBoost, while Random Forest also provides competitive results.

4.1.1. Comparison of results with recent models

In this analysis, we compare our results on S&P 500 stock index predictions with findings from three distinct studies: Gao et al. (2017), Sarıkoç and Celik (2024), and Hossain et al. (2018). This comparison spans several key metrics, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE), which illustrate the effectiveness of various machine learning models in financial forecasting (Table 4).

As shown in the Table 4, the current study employing Long Short-Term Memory (LSTM) without lag demonstrates exceptional performance with a Mean Absolute Error (MAE) of 0.014, a Mean Absolute Percentage Error (MAPE) of 0.008, and a Root Mean Square Error (RMSE) of 0.045. This positions it as the most accurate model in the comparison. In contrast, Hossain et al. (2018) introduced a hybrid approach utilizing LSTM and Gated Recurrent Units (LSTM+GRU), achieving a slightly higher MAE of 0.023 and MAPE of 0.0413, yet with a notably low RMSE of 0.0023.

In addition, Hossain et al. also presented the PCA-ICA-LSTM model, which yielded an MAE of 0.0147 and a MAPE of 0.0205 with an even smaller RMSE of 0.00038, showcasing its effectiveness in capturing market trends. Sarıkoç and Celik (2024) further utilized PCA-ICA-LSTM, recording an MAE of 0.077 and MAPE of 0.042, alongside an RMSE of 0.084, indicating solid performance but not quite as effective as the previous models. In stark contrast, Gao et al. (2017) reported significantly poorer results with their LSTM model, revealing an MAE of 14.7709, MAPE of 0.7240, and RMSE of 20.4668, highlighting the challenges encountered in traditional LSTM approaches compared to the more advanced methodologies adopted in recent studies.

Overall, these results highlight significant advancements in machine learning techniques for financial forecasting. Specifically, hybrid models combined with supportive variables like technical indicators have shown significantly enhanced accuracy compared to traditional LSTM implementations in this research.

4.2. Feature importance

As previously mentioned, the primary objective of this research is to identify key technical indicators that can be used to predict the S&P 500 index. In this section, we present the results from our experiments which focused on determining the most significant technical indicators for predicting the S&P 500 index. Please note that certain features such as close, high, open, and low prices were included in all experiments. However, they are not specifically mentioned in the following results as they do not fall into any of the indicator categories.

As shown in Fig. 4, MFI is the best indicator for SVR without lag and HLC3 is the best indicator for the SVR with lag in Close price. TEMA, Volume, FWMA, and OBV are the next most important indicators for SVR without lag and SVR with lag in Close price.

Fig. 5 illustrates that Ichimuku in XGBoost with no lag in Close price and HWMA with lag in Close price are the most important indicators followed by OBV, TEMA, SMA14, and HLC3. It is noticeable that OBV, HLC3, and TEMA also appeared in the SVR method as the best indicators for the prediction of the S&P 500 index.

Using Random Forest with no lag in Close price, OBV, and with lag in Close price, HLC3 are the most important indicators as shown in Fig. 6. Furthermore, SWMA, PVT, BBL, and TEMA are selected afterward. It is also clear that, OBV, HLC3. PVT, Ichimuku, and FWMA have appeared in top selections of either SVR or XGBoost techniques.

Finally, Fig. 7 shows that HLC3 is chosen as the most important indicator for LSTM (with and without price lag), followed by OBV, SMA5, TEMA, BBL, and SMA50 in the top 10 selected indicators. HLC3, OBV, TEMA, and BBL also appeared in previous methods.

In addition, we have summarized the outcome of the selected indicators for all machine learning (ML) models in the following table . The table has marked indicators that are selected for each Machine learning (ML) model, (Table 5).

Momentum Indicators: Indicators like KAMA, RVI, SMA, and HLC3 are utilized across various models, with SMA being particularly common. These metrics help quantify the rate of price change, assisting in identifying overbought or oversold conditions.

Trend Indicators: Indicators such as MI, EMA, TEMA, HMA, and Ichimoku show varied selections among the models, highlighting their importance in trend analysis. They are used to determine market direction and reduce noise.

Volatility Indicators: ATR, BB, and Price Distance measure price dispersion and are included in most models. They are essential for assessing risk and detecting potential breakouts.

Volume Indicators: MFI, OBV, and PVT are integrated into different

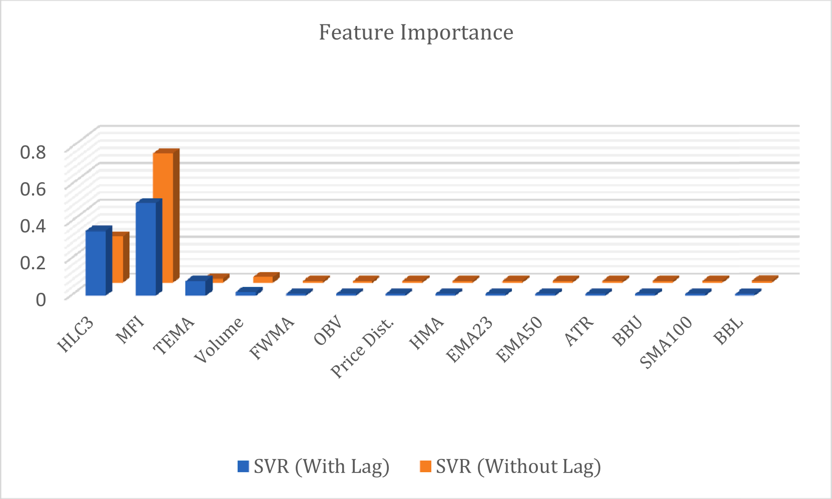

Fig. 4. Feature important results for SVR.
Note: The top 2 indicators (MFI, HLC3) explain >80% of the variance in SVR models

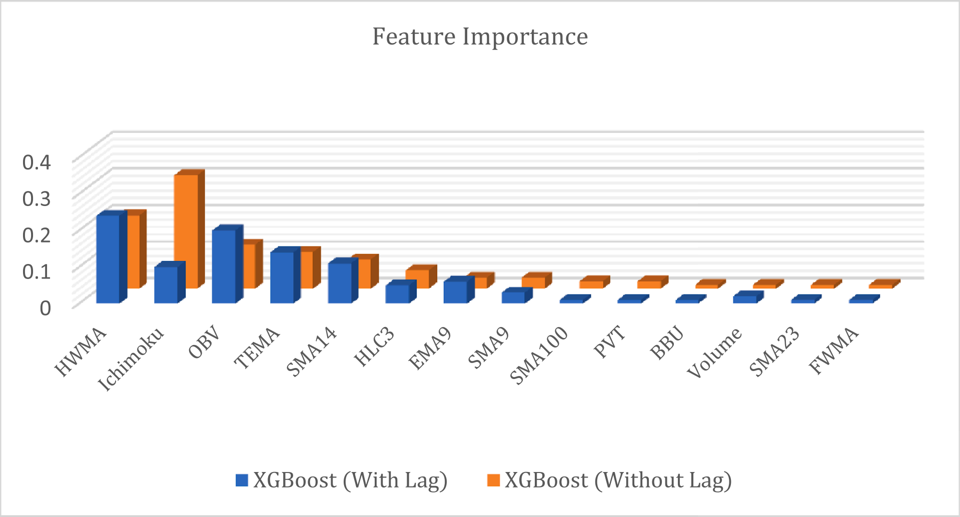

Fig. 5. Feature important results for XGBoost.

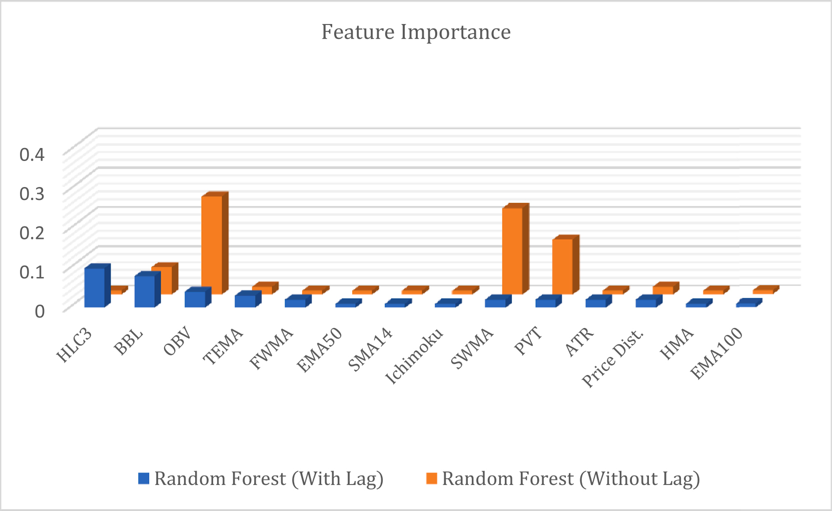

Fig. 6. Feature important results for random forest.

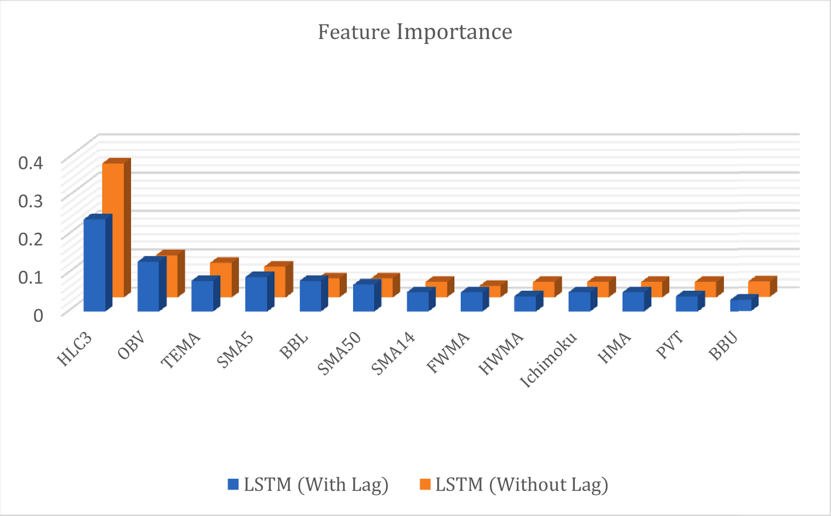

Fig. 7. Feature important results for LSTM.

Table 4
Comparison of research results with recent models.

S&P 500 Forecasting	MAE	MAPE	RMSE
Current Study (LSTM without lag in price)	0.014	0.008	0.045
Hossain et al. (2018) Proposed LSTM+GRU	0.023	0.0413	0.0023
Hossain et al. (2018) PCA-ICA-LSTM	0.0147	0.0205	0.00038
Sarıkoç & Celik (2024) PCA-ICA-LSTM	0.077	0.042	0.084
Gao et al. (2017) LSTM	14.7709	0.7240	20.4668

models, emphasizing the role of trading volume in market predictions. These indicators reflect underlying market sentiment and trends in accumulation or distribution.

Fig. 8 represents the most frequent and influential indicators in each indicator category. These categories are momentum, trend, volatility, and volume. As can be seen in this figure, RVI, KAMA, SWMA, and HLC3 are selected in the momentum category as the most important ones. Also, the most essential trend indicators are MI, EMA, TEMA, FWMA, HMA, SWMA, HWMA, and Ichimuku Cloud. In the volatility class, ATR, Price Distance, and Bolinger Bands, and in the volume class, MFI, OBV, and PVT were identified as the most influential ones.

5. Discussion and limitations

First, it is essential to recognize that while the overarching trend of the S&P 500's closing prices shows a consistent upward path, the primary contribution of this study is to improve short-term prediction accuracy, particularly for daily forecasts. The long-term growth of the market is a well-documented phenomenon; however, the real challenge lies in making precise predictions within narrower timeframes, such as daily, weekly, monthly, or yearly intervals. This study aims to address that challenge by identifying and utilizing key technical indicators that can enhance prediction accuracy in these shorter timeframes.

To further this goal, we have emphasized the importance of focusing on short-term prediction accuracy and proposed the exploration of specific indicators in future work. These indicators can play a crucial role in refining forecasts for intra-week and intra-month trading strategies, ultimately contributing to more effective decision-making in volatile market conditions. By concentrating on this aspect, the study aims to provide actionable insights that go beyond general market

Table 5
Selection of technical indicators in each machine learning (ML) model.

	XGBoost No Lag	XGBoost With Lag	Random Forest No Lag	Random Forest With Lag	SVR No Lag	SVR With Lag	LSTM No Lag	LSTM With Lag
Momentum Indicators								
KAMA			√	√	√	√		
RVI	√							
SMA	√	√	√	√	√	√		
HLC3	√	√	√	√	√	√	√	√
Trend Indicators								
MI	√							
EMA	√	√	√	√	√	√		
TEMA	√	√	√	√	√	√	√	√
FWMA	√	√	√	√	√	√	√	√
HMA			√	√	√	√	√	√
SWMA			√	√			√	√
HWMA	√	√			√	√	√	√
Ichimoku	√	√	√	√			√	√
Volatility Indicators								
ATR			√	√	√	√		
BB	√	√	√	√	√	√	√	√
Price Distance			√	√	√	√		
Volume Indicators								
MFI					√	√		
OBV	√	√	√	√	√	√	√	√
PVT	√	√	√	√			√	√

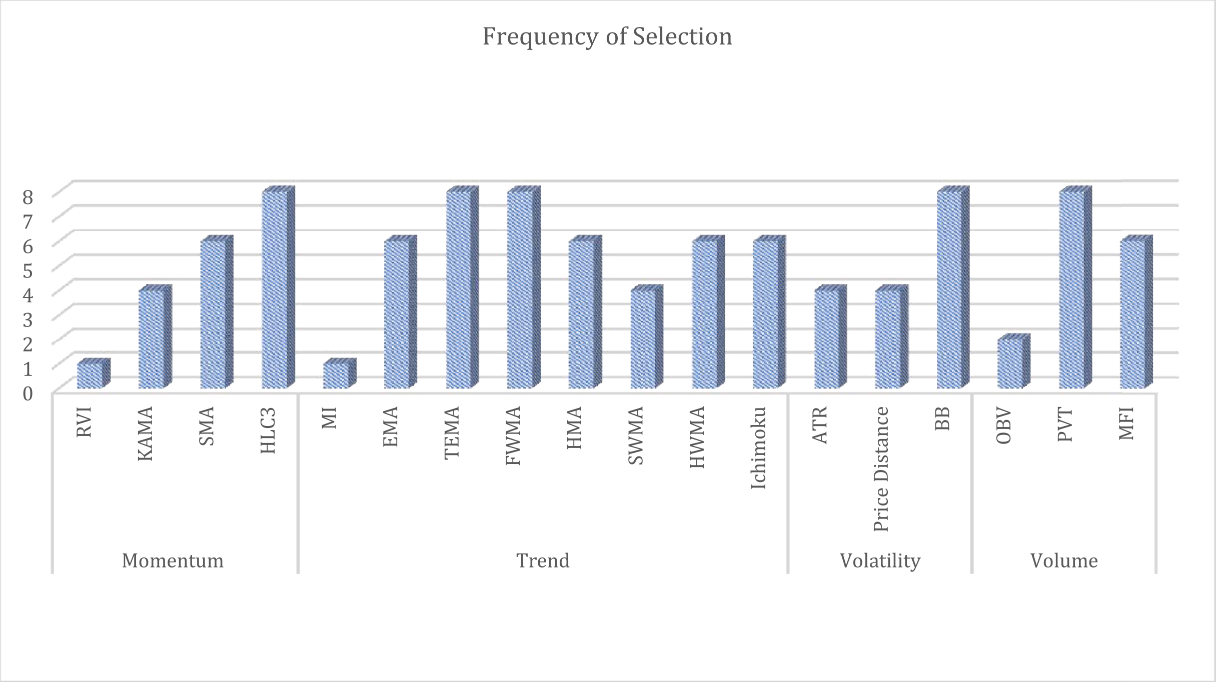

Fig. 8. Most important indicators for each category.

trends, offering value to traders and analysts seeking to navigate short-term fluctuations.

Through the application of multiple regression models, we were able to uncover the most significant indicators across various categories, including momentum, trend, volatility, and volume. Key findings from our analysis include:

- **Momentum** Indicators**:** RVI, KAMA, SWMA, and HLC3 were identified as the most significant momentum indicators.
- **Trend** Indicators**:** MI, EMA, TEMA, FWMA, HMA, SWMA, HWMA, and Ichimoku cloud were found to be important trend indicators.
- **Volatility** Indicators**:** ATR, Price Distance, and Bollinger Bands were identified as key volatility indicators.
- **Volume** Indicators**:** MFI, OBV, and PVT were shown to be significant volume indicators.

By focusing on the most informative indicators, we can reduce the complexity of our models and improve their predictive accuracy.

Also, Section 4.2 reveals notable divergences in feature importance rankings across the evaluated machine learning models, attributable to their inherent algorithmic architectures and inductive biases. These distinctions are elucidated as follows:

Long Short-Term Memory (LSTM) Models: LSTM architectures exhibit a pronounced emphasis on temporal dependencies, prioritizing time-sensitive indicators such as HLC3 (a weighted average of High, Low, and Close prices) and SMA5 (5-day Simple Moving Average). This aligns with their capacity to model sequential patterns, capturing how price dynamics evolve over discrete intervals.

Tree-Based Models (XGBoost and Random Forest): These models leverage their non-parametric structure to emphasize OBV (On-Balance Volume), reflecting their proficiency in disentangling complex, nonlinear interactions between trading volume and price fluctuations. The hierarchical decision-making process inherent to tree ensembles enables robust identification of volume-driven market signals.

Support Vector Regression (SVR) Models: SVR frameworks derive predictive utility from smoothed trend indicators, such as TEMA (Triple Exponential Moving Average) and EMA23 (23-day Exponential Moving Average). Their reliance on kernel-based optimization favors features with reduced noise, aligning with the mathematical prerequisites for effective hyperplane construction in high-dimensional spaces.

These disparities in feature prioritization underscore the heterogeneous interpretative frameworks adopted by distinct model classes. As posited in Section 3, this heterogeneity necessitates the development of generalized feature selection methodologies adaptable to diverse algorithmic paradigms. Furthermore, recognizing the unique inductive biases of each model—whether temporal sensitivity (LSTM), nonlinear volume interactions (tree-based models), or smoothed trend adherence (SVR)—enables strategic alignment between indicator selection and model-specific strengths. Such alignment is critical for optimizing predictive accuracy and ensuring robustness across varying market conditions.

While our research provides valuable insights into the selection of key technical indicators, it is important to acknowledge certain limitations:

Data Limitations: While the dataset spans 70 years, which offers extensive historical insights, it may also pose challenges. Over time, markets undergo structural changes due to factors like regulatory shifts, technological innovations, and economic fluctuations. Indicators based on older data might not accurately reflect current market conditions, such as the rise of algorithmic trading or major events like the 2020 pandemic. This can limit the model's ability to generalize effectively in today's market. Furthermore, generalizability to other indices (e.g., Dow Jones, Nasdaq) or international markets (e.g., FTSE 100) is untested, and future research will validate the framework on broader datasets.

Market Dynamics: The stock market is a complex system influenced by numerous factors, including economic conditions, geopolitical events, and investor sentiment. Our models may not be able to capture all of these factors, potentially limiting their predictive power.

Model Limitations: We evaluated several models (XGBoost, Random Forest, SVR, LSTM), but we didn't explore their specific strengths and weaknesses in detail. For example, LSTMs are great for understanding time-based patterns but need more computational power and careful tuning. On the other hand, SVR's success depends on choosing the right kernel for capturing non-linear trends. Discussing these trade-offs could help clarify why certain models were chosen.

Overfitting: Although we took steps to reduce overfitting—like using PCA for dimensionality reduction, setting dropout layers in LSTM (0.2), and applying XGBoost regularization ($\lambda = 0.5$)—the complexity of models like LSTM still presents risks. Financial data is often noisy, which makes overfitting more likely. We used walk-forward validation to address this, but relying solely on past patterns assumes that market conditions remain stable, which might not be the case during significant changes or crises.

6. Conclusion

This paper focuses on identifying key indicators for predicting the S&P 500 stock index. Multiple regression models were used to uncover the most important technical indicators, or features. To capture these indicators, various techniques were employed, including XGBoost, Random Forest, Support Vector Regression, and LSTM regression. The input data consisted of 88 technical indicators after applying the PCA method for dimension reduction. In this research, all the models were applied twice. First, using the Close price of the current day as the dependent variable, and second, with a one-day lag in the Close price. The results reveal the most significant technical indicators across different categories, including momentum, trend, volatility, and volume. The following section provides a summary of the selected key indicators, obtained through different machine learning models, both with and without lag in the Close price.

- For SVR without lag in Close price MFI, and the SVR with lag in Close price, HLC3 are the best-performing indicators.
- In XGBoost with no lag in Close price, Ichimuku and with lag in Close price HWMA are the best-performing indicators.
- Using Random Forest with no lag in Close price, OBV, and with lag in Close price, HLC3 are the most important indicators.
- HLC3 is selected as the best-performing indicator for LSTM, with and without lag in Close price.
- It is indicated that HLC3, TEMA, FWMA, OBV, and Bollinger Bands appeared in all models.

Also, in the case of indicator categories of momentum, trend, volatility, and volume, the following indicators appear as the best-performing results:

- In the category of momentum indicators, RVI, KAMA, SWMA, and HLC3
- In the category of trend indicators, MI, EMA, TEMA, FWMA, HMA, SWMA, HWMA and Ichimoku cloud
- In the category of volatility indicators, ATR, Price Distance, and Bolinger Bands
- In the category of volume indicators, MFI, OBV, and PVT

From this, we can conclude that selecting only the required number of technical indicators with a strong effect on the target variable results in the least error metrics. Future research could explore the following areas:

- Expanding the dataset: Incorporating additional data sources, such as news sentiment or economic indicators, could improve the predictive power of the models.
- Testing the models on different markets: Applying our methodology to other stock indices or asset classes can help assess its generalizability.
- Developing hybrid models: Combining machine learning techniques with traditional technical analysis methods could provide new insights and improve prediction accuracy.
- Investigating the impact of time horizons: Analyzing the performance of the models over different time horizons (e.g., short-term, medium-term, long-term) can provide valuable information for investors and traders.
- Enhancing model interpretability: Developing methods to improve the interpretability of the forecasting models will help users understand how specific indicators influence predictions.

By addressing these limitations and exploring future research directions, we can continue to advance our understanding of technical indicators and their role in stock market prediction.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

List of all Technical Indicators

1	Accumulation Distribution	31	Volume	61	KST Indicator
2	ADX	32	Volume Oscillator	62	Linear Regression
3	Aroon Oscillator	33	Weighted Close	63	Linear Regression Indicator
4	ATR Bands	34	Weighted Moving Average	64	The Moving Average Oscillator
5	ATR Trailing Stops	35	Wilder Moving Average	65	MACD Indicator
6	Average True Range	36	Williams %R	66	MACD Histogram
7	Bollinger Bands	37	Williams Accumulate Distribute	67	MACD Percentage
8	Bollinger Band® Width	38	Williams Accumulation Distribution	68	Mass Index
9	Bollinger %B	39	Chande Momentum Oscillator	69	Median Price
10	Candlestick Patterns	40	Chandelier Exits	70	Momentum Indicator
11	Chaikin Money Flow	41	Choppiness Index	71	Money Flow Index
12	Chaikin Oscillator	42	Commodity Channel Index	72	Moving Average
13	Chaikin Volatility	43	Compare Prices	73	Moving Average Filters
14	Rate of Change (Volume)	44	Coppock Indicator	74	Moving Average High/Low/Open
15	Relative Strength (Compare)	45	Detrended Price Oscillator	75	Negative Volume
16	Relative Strength Index (RSI)	46	Directional Movement	76	On Balance Volume
17	Safezone Indicator	47	Displaced Moving Average	77	Parabolic SAR
18	Simple Moving Average	48	Donchian Channels	78	Percentage Bands
19	Standard Deviation Channels	49	Ease of Movement	79	Percentage Trailing Stops
20	Stochastic Oscillator	50	Elder Ray Index	80	Pivot Points

(*continued on next page*)

(*continued*)

21	Stochastic RSI	51	Fibonacci Extensions	81	Positive Volume
22	Trend Lines	52	Fibonacci Retracements	82	Price Comparison
23	TRIX Indicator	53	Force Index	83	Price Differential
24	True Range	54	Heikin-Ashi Candlesticks	84	Price Envelope
25	Typical Price	55	Hull Moving Average	85	Price Ratio
26	Ultimate Oscillator	56	Ichimoku Cloud	86	Price Volume Trend
27	Vertical Horizontal Filter (VHF)	57	Inverted Axis	87	Rainbow 3D Moving Averages
28	Volatility	58	Keltner Channels	88	Rate of Change (Price)
29	Volatility Ratio	59	Exponential Moving Average		
30	Volatility Stops	60	Multiple Moving Averages		

Data availability

Data will be made available on request.

References

Ahn, D. H., Conrad, J., & Dittmar, R. F. (2003). Risk adjustment and trading strategies. *The Review of Financial Studies, 16*(2), 459–485.

Alsubaie, Y., Hindi, K. E., & Alsalman, H. (2019). Cost-sensitive prediction of stock price direction: Selection of technical indicators. *IEEE Access, 7*, 146876–146892. https://doi.org/10.1109/ACCESS.2019.2945907

Ampomah, Ernest Kwame, Qin, Zhiguang, Nyame, Gabriel, & Botchey, Francis Effirm (2021). Stock market decision support modeling with tree-based AdaBoost ensemble machine learning models. *Informatica, 44*(4).

Ayitey Junior, M., Appiahene, P., Appiah, O., & Bombie, C. N. (2023). Forex market forecasting using machine learning: Systematic Literature Review and meta-analysis. *Journal of Big Data, 10*(1), 9.

Bâra, A., & Oprea, S. V. (2024). An ensemble learning method for Bitcoin price prediction based on volatility indicators and trend. *Engineering Applications of Artificial Intelligence, 133*, Article 107991.

Basak, Suryoday, Kar, Saibal, Saha, Snehanshu, Khaidem, Luckyson, & Dey, Sudeepa Roy (2019). Predicting the direction of stock market prices using tree-based classifiers. *The North American Journal of Economics and Finance, 47*, 552–567. https://doi.org/10.1016/j.najef.2018.06.013

Board, Financial Stability. (2017). *Artificial intelligence and machine learning in financial services*. November.

Botunac, Ive, Panjkota, Ante, & Matetic, Maja (2020). The effect of feature selection on the performance of long short-term memory neural network in stock market predictions. In *31st DAAAM ISIMA* (pp. 592–598).

Chandar, S. Kumar (2022). Convolutional neural network for stock trading using technical indicators. *Automated Software Engineering, 29*, 1–14.

Chen, Yingjun, & Hao, Yongtao (2017). A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction. *Expert Systems with Applications, 80*, 340–355. https://doi.org/10.1016/j.eswa.2017.02.044

Chung, Hyejung, & Shin, Kyung Shik (2018). Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability, 10*(10), 3765. https://doi.org/10.3390/SU10103765. *2018, Vol. 10, Page 3765*.

Fang, Jiali, Qin, Yafeng, & Jacobsen, Ben (2014). Technical market indicators: An overview. *Journal of Behavioral and Experimental Finance, 4*, 25–56.

Fernández, María Ferrer, Henry, Ólan, Pybis, Sam, & Stamatogiannis, Michalis P. (2023). Can we forecast better in periods of low uncertainty? The role of technical indicators. *Journal of Empirical Finance, 71*, 1–12.

Gao, T., Chai, Y., & Liu, Y. (2017). Applying long short term momory neural networks for predicting stock closing price. In *2017 8th IEEE international conference on software engineering and service science (ICSESS)*.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). Cambridge: MIT press (No. 2).

Gorenc Novak, Marija, & Velušček, Dejan (2016). Prediction of stock price movement based on daily high prices. *Quantitative Finance, 16*(5), 793–826. https://doi.org/10.1080/14697688.2015.1070960

Van der Hagen, R. (2021). *Predicting Bitcoin price using technical indicator data features in Long short-term Memory models*. Tilburg University.

Han, Yechan, Kim, Jaeyun, & Enke, David (2023). A machine learning trading system for the stock market based on N-period min-max labeling using XGBoost. *Expert Systems with Applications, 211*, Article 118581.

Hossain, M. A., Karim, R., Thulasiram, R., Bruce, N. D., & Wang, Y. (2018). Hybrid deep learning model for stock price prediction. In *2018 IEEE symposium series on computational intelligence (SSCI)* (pp. 1837–1844). IEEE.

Htun, H. H., Biehl, M., & Petkov, N. (2023). Survey of feature selection and extraction techniques for stock market prediction. *Financial Innovation, 9*(1), 26.

Jiang, Z., Ji, R., & Chang, K. C. (2020). A machine learning integrated portfolio rebalance framework with risk-aversion adjustment. *Journal of Risk and Financial Management, 13*(7), 155.

Ji, Gang, Yu, Jingmin, Hu, Kai, Xie, Jie, & Ji, Xunsheng (2022). An adaptive feature selection schema using improved technical indicators for predicting stock price movements. *Expert Systems with Applications, 200*, Article 116941.

Jin, Haifeng, Chollet, François, Song, Qingquan, & Hu, Xia (2023). Autokeras: An Automl library for deep learning. *Journal of Machine Learning Research, 24*(6), 1–6.

Ku, Chin Soon, Xiong, Jiale, Chen, Yen-Lin, Cheah, Shing Dhee, Soong, Hoong Cheng, & Por, Lip Yee (2023). Improving stock market predictions: An equity forecasting scanner using long short-term memory method with dynamic indicators for Malaysia stock market. *Mathematics, 11*(11), 2470.

Kuber, Vishal, Divakar Yadav, and Arun Kr Yadav. 2022. "Univariate and multivariate LSTM model for short-term stock market prediction".

Kumar, Deepak, Meghwani, Suraj S., & Thakur, Manoj (2016). Proximal support vector machine based hybrid prediction models for trend forecasting in financial markets. *Journal of Computational Science, 17*, 1–13. https://doi.org/10.1016/j.jocs.2016.07.006

Kumbure, Mahinda Mailagaha, Lohrmann, Christoph, Luukka, Pasi, & Porras, Jari (2022). Machine learning techniques and data for stock market forecasting: A literature review. *Expert Systems with Applications, 197*, Article 116659.

Nabipour, Mojtaba, Nayyeri, Pooyan, Jabani, Hamed, Shahab, S., & Mosavi, Amir (2020). Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; A comparative analysis. *IEEE Access, 8*, 150199–150212. https://doi.org/10.1109/ACCESS.2020.3015966

Naik, Nagaraj, & Mohan, Biju R. (2019). Stock price movements classification using machine and deep learning techniques-The case study of Indian stock market. In J. Macintyre, L. Iliadis, I. Maglogiannis, & C. Jayne (Eds.), *Engineering Applications of Neural Networks* (pp. 445–452). Cham: Springer International Publishing.

Nazareth, N., & Reddy, Y. V. R. (2023). Financial applications of machine learning: A literature review. *Expert Systems with Applications, 219*, Article 119640.

Ntakaris, Adamantios, Mirone, Giorgio, Kanniainen, Juho, & Moncef Gabbouj, and Alexandros Iosifidis. (2019). Feature engineering for mid-price prediction with deep learning. *IEEE Access, 7*, 82390–82412. https://doi.org/10.1109/ACCESS.2019.2924353

Obthong, M., Tantisantiwong, N., & Jeamwatthanachai, W. (2020). *A survey on machine learning for stock price prediction: Algorithms and techniques*.

Park, Hyun Jun, Kim, Youngjun, & Kim, Ha Young (2022). Stock market forecasting using a multi-task approach integrating long short-term memory and the random forest framework. *Applied Soft Computing, 114*, Article 108106.

Patel, Jigar, Shah, Sahil, Thakkar, Priyank, & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications, 42*(1), 259–268. https://doi.org/10.1016/j.eswa.2014.07.040

Patel, Jigar, Shah, Sahil, Thakkar, Priyank, & Kotecha, Ketan (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications, 42*(4), 2162–2172.

Peng, Yaohao, Albuquerque, Pedro Henrique Melo, Kimura, Herbert, & Saavedra, Cayan Atreio Portela Bárcena (2021). Feature selection and deep neural networks for stock price direction forecasting using technical analysis indicators. *Machine Learning with Applications, 5*, Article 100060.

Pérez-Hernández, F., & Arévalo-de-Pablos, A. (2024). A hybrid model integrating artificial neural network with multiple GARCH-type models and EWMA for performing the optimal volatility forecasting of market risk factors. *Expert Systems with Applications, 243*, Article 122896.

Phillips, Peter C. B., & Shi, Shuping (2020). Real time monitoring of asset markets: Bubbles and crises. In *Handbook of Statistics, 42* pp. 61–80). Elsevier.

Prasad, Akhilesh, & Bakhshi, Priti (2022). Forecasting the direction of daily changes in the India VIX index using machine learning. *Journal of Risk and Financial Management, 15*(12), 552.

Qiu, Mingyue, & Song, Yu (2016). Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLoS One, 11*(5), Article e0155133. https://doi.org/10.1371/JOURNAL.PONE.0155133

Rundo, F., Trenta, F., di Stallo, A. L., & S. Battiato-Applied Sciences, and undefined 2019. (2019). *Machine learning for quantitative finance applications: A survey*. Mdpi.Com. https://doi.org/10.3390/app9245574

Sakhare, Nitin Nandkumar, Shaik, Imambi S., & Saha, Suman (2023). Prediction of stock market movement via technical analysis of stock data stored on blockchain using novel history bits based machine learning algorithm. *IET Software*.

Salim, M., & Djunaidy, A. (2024). Development of a CNN-LSTM approach with images as time-series data representation for predicting gold prices. *Procedia Computer Science, 234*, 333–340.

Sarıkoç, M., & Celik, M. (2024). PCA-ICA-LSTM: A hybrid deep learning model based on dimension reduction methods to predict S&P 500 index price. *Computational Economics*, 1–67.

Smilevski, M. (2020). Applying recent advances in visual question answering to record linkage. arXiv preprint arXiv:2007.05881.

Song, Yoojeong, Lee, Jae Won, & Lee, Jongwoo (2019). A study on novel filtering and relationship between input-features and target-vectors in a deep learning model for stock price prediction. *Applied Intelligence, 49*(3), 897–911. https://doi.org/10.1007/S10489-018-1308-X/METRICS

Weng, Bin, Lu, Lin, Wang, Xing, Megahed, Fadel M., & Martinez, Waldyn (2018). Predicting short-term stock prices using ensemble methods and online data sources. *Expert Systems with Applications, 112*, 258–273. https://doi.org/10.1016/j.eswa.2018.06.016

Yin, Lili, Li, Benling, Li, Peng, & Zhang, Rubo (2023). Research on stock trend prediction method based on optimized random forest. *CAAI Transactions on Intelligence Technology, 8*(1), 274–284.

Yuan, Xianghui, Yuan, Jin, Jiang, Tianzhao, & Ul Ain, Qurat (2020). Integrated long-term stock selection models based on feature selection and machine learning algorithms for china stock market. *IEEE Access, 8*, 22672–22685. https://doi.org/10.1109/ACCESS.2020.2969293

Yun, Kyung Keun, Yoon, Sang Won, & Won, Daehan (2021). Prediction of stock price direction using a hybrid GA-XGBoost algorithm with a three-stage feature engineering process. *Expert Systems with Applications, 186*, Article 115716.

Zhang, C., Sjarif, N. N. A., & Ibrahim, R. (2024). Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 14*(1), e1519.

Zhao, J., Ouenniche, J., & De Smedt, J. (2024). Survey, classification and critical analysis of the literature on corporate bankruptcy and financial distress prediction. *Machine Learning with Applications*, Article 100527.

Zhong, Shan, and David B. Hitchcock. 2021. "S&P 500 stock price prediction using technical, fundamental and text data." *arXiv Preprint arXiv:2108.10826*.