ARTICLE IN PRESS

Borsa Istanbul Review xxx (xxxx) xxx

Contents lists available at ScienceDirect

Borsa Istanbul Review

journal homepage: www.elsevier.com/journals/borsa-istanbul-review/2214-8450

Stock price prediction using the Sand Cat Swarm Optimization and an improved deep Long Short Term Memory network

Burak Gülmez [a,b,*]

[a] Department of Industrial Engineering, Mudanya University, 16940, Mudanya, Bursa, Türkiye
[b] Leiden Institute of Advanced Computer Science, Leiden University, Leiden, Netherlands

ARTICLE INFO

Keywords:
Stock price prediction
Sand Cat swarm optimization
LSTM
Deep learning
Artificial intelligence in finance
Financial forecasting

ABSTRACT

Stock price prediction remains a complex challenge in financial markets. This study introduces a novel Long Short-Term Memory (LSTM) model optimized by Sand Cat Swarm Optimization (SCSO) for stock price prediction. The research evaluates multiple algorithms including ANN, LSTM variants, Auto-ARIMA, Gradient Boosted Trees, DeepAR, N-BEATS, N-HITS, and the proposed LSTM-SCSO using DAX index data from 2018 to 2023. Model performance was assessed through Mean Squared Error, Mean Absolute Error, Mean Absolute Percentage Error, and out-of-sample R2 metrics. Statistical significance was validated using Model Confidence Set analysis with 5000 bootstrap replications. Results demonstrate LSTM-SCSO's superior performance across all evaluation metrics. The model achieved an annualized return of 66.25% compared to the DAX index's 47.45%, with a Sharpe ratio of 2.9091. The integration of technical indicators and macroeconomic variables enhanced the model's predictive capabilities. These findings establish LSTM-SCSO as an effective tool for stock price prediction, offering practical value for investment decision-making.

1. Introduction

Stock price prediction is a critical yet challenging task in financial markets. The complexity arises from the multitude of factors influencing stock prices, including economic indicators, company performance, market sentiment, and global events. Accurate predictions can significantly impact investment strategies, risk management, and financial decision-making processes. The stock market's inherent volatility and sensitivity to various external factors make it a complex system to analyze and predict. Traditional forecasting methods often struggle to capture the non-linear relationships and temporal dependencies present in stock price data. This complexity necessitates the development of more sophisticated prediction models that can adapt to changing market conditions and capture intricate patterns in financial time series. (Hellström and Holmström, 1998; Kumar & Dadhich, 2014).

The field of artificial intelligence (AI) is currently experiencing significant popularity, and there are numerous studies that have been conducted on the subject (Gülmez, 2023c, 2023b, 2023a, 2024a, 2024b, 2024c, 2024d; Gülmez, Emmerich, & Fan, 2024; Gülmez & Kulluk, 2019, 2023a). AI and machine learning techniques have emerged as powerful tools for addressing the challenges of stock price prediction. These methods can analyze vast amounts of data, identify complex patterns, and make predictions based on historical trends and current market conditions. Among these techniques, Long Short-Term Memory (LSTM) networks have shown particular promise due to their ability to capture long-term dependencies in sequential data, making them well-suited for time series forecasting tasks such as stock price prediction (Chopra & Sharma, 2021; M et al., 2022).

Time series analysis is a popular technique used to analyze and predict stock prices. One common approach is to use ANNs such as recurrent neural networks (RNNs) and LSTM networks. ANNs can process large amounts of data and identify patterns that might be difficult for humans to detect. RNNs are particularly useful for time series analysis because they can consider the sequence of data points and learn from past trends to make future predictions. LSTM networks are a type of RNN that are designed to handle long-term dependencies and are especially useful for predicting stock prices over longer periods.

In this paper, a novel LSTM model optimized by SCSO Algorithm (Seyyedabbasi & Kiani, 2023) is proposed to predict stock prices. The LSTM model is a type of recurrent neural network that can learn from and make predictions on sequential data, such as time series data. The SCSO Algorithm is a meta-heuristic optimization algorithm that is used

* Corresponding author. Department of Industrial Engineering, Mudanya University, 16940, Mudanya, Bursa, Türkiye.
E-mail addresses: b.gulmez@liacs.leidenuniv.nl, burak.gulmez@mudanya.edu.tr.

https://doi.org/10.1016/j.bir.2024.12.002
Received 9 August 2024; Received in revised form 17 December 2024; Accepted 18 December 2024
Available online 19 December 2024

to optimize the LSTM model. The resulting model is able to make accurate predictions of stock prices, taking into account historical data, market trends, and other relevant factors. By using this model, traders and investors can make more informed decisions about when to buy and sell stocks, potentially leading to increased profits and reduced risk. This paper proposes a new deep-learning model optimized by SCSO to predict stock prices.

The primary contributions and innovations of this study are as follows:

1. Novel LSTM-SCSO Model: This paper introduces a new deep-learning model that combines LSTM networks with the SCSO algorithm for stock price prediction. This is the first application of SCSO in optimizing LSTM for financial forecasting, representing a significant advancement in the field.
2. Comprehensive Performance Comparison: The study provides a thorough comparison of the proposed LSTM-SCSO model against other popular algorithms (ANN, LSTM1D, LSTM2D, LSTM3D, LSTM-GA, Auto-ARIMA, GBT, DeepAR, N-BEATS, N-HITS) using multiple evaluation metrics. This comprehensive analysis offers insights into the relative strengths and weaknesses of different approaches to stock price prediction.
3. Improved Prediction Accuracy: The LSTM-SCSO model demonstrates superior performance across various stock tickers, addressing the challenge of accurate stock price forecasting. This improvement in prediction accuracy can lead to more informed investment decisions and better risk management strategies.
4. Adaptive Optimization: By incorporating SCSO, the model can adaptively optimize LSTM parameters, potentially capturing market dynamics more effectively than traditional optimization methods. This adaptive approach addresses the challenge of model flexibility in the face of changing market conditions.
5. Practical Application to DAX Index: The study applies the proposed model to real-world data from the DAX index, demonstrating its practical utility in a major financial market. This application provides valuable insights for investors and traders operating in the German stock market.

By addressing these points, this research contributes to the ongoing efforts to improve the accuracy and reliability of stock price predictions, ultimately aiming to enhance decision-making processes in financial markets.

2. Literature review

2.1. Traditional time series analysis methods

Traditional time series analysis methods have long been employed in stock market prediction. These methods typically rely on historical data to forecast future trends.

Duan et al. (2021) proposed a non-linear delay grey prediction model using impulse delay differential equations and particle swarm optimization, demonstrating superior performance compared to conventional grey and time series models. Xiao and Su (2022) applied the ARIMA (autoregressive integrated moving average) model to predict stock prices, comparing it with deep learning methods. Maguluri and Ragupathy (2020) used the ARIMA model for future price prediction and forecasting in stock markets. Almaafi et al. (2023) evaluated and compared the performance of ARIMA and XGBoost models in predicting weekly closing prices of Saudi Telecom Company stock. Zhang et al. (2022) employed a mixed model combining random walk and time series approaches for stock price forecasting. Li and Abo Keir (2022) proposed a Stock Price Prediction mathematical Model (SPPM) based on BP neural network with high-frequency data. Nakagawa et al. (2019) introduced a stock price prediction method using k-medoid clustering with indexing dynamic time warping. Ilyas et al. (2022) proposed a hybrid model using a fully modified Hodrick–Prescott filter for noise reduction in stock price data. Fujimoto et al. (2022) presented the Uncertainty Aware Trader-Company technique (UTC) to combine the TC technique with probabilistic modeling for probabilistic forecasts and uncertainty estimations.

2.2. Machine learning approaches

Machine learning algorithms have gained significant traction in stock price prediction due to their ability to handle complex, non-linear relationships in financial data.

Basak et al. (2019) used random forests and gradient-boosted decision trees (XGBoost) to predict the direction of stock market prices, achieving high accuracy for medium to long-run predictions. Bazrkar and Hosseini (2023) utilized PSO parameter adjustment to enhance stock market prediction using a Support Vector Machine (SVM). Dwivedi and Gore (2021) proposed a historical data-based ensemble system combining various machine learning-based prediction models using LASSO regression regularization. Jafari and Haratizadeh (2022) introduced GCNET, a graph-based prediction model for stock price movement using graph convolutional networks. G. Li et al. (2022) evaluated stock market forecasting frameworks for AI and embedded real-time systems, focusing on neural network prediction methods. Staffini (2022) introduced a Deep Convolutional Generative Adversarial Network (DCGAN) architecture for forecasting the closing price of stocks. Rajanikanth, Haritha, and Shankar (2023) used Linear Regression (LR) to develop a new Stock Close Price Prediction (SCPP) algorithm. Tang et al. (2020) developed forecasting models with deep learning technology for share price prediction in the logistics industry. Vanstone et al. (2019) incorporated sentiment predictors based on news articles and Twitter posts to improve stock price predictions. Yu et al. (2020) proposed a stock price forecasting model based on the LLE-BP neural network. Mian (2023) evaluated stock closing prices using Transformer learning, comparing it with ARIMA, LSTM, and Random Forest algorithms. Wang et al. (2022) developed a stock price forecasting model based on wavelet filtering and an ensembled machine learning approach.

2.3. Deep learning models

Deep learning models, particularly those based on neural networks, have become increasingly popular for stock price prediction due to their ability to capture complex temporal dependencies.

Yuan et al. (2023) developed a PSO-LSTM stock forecasting model to improve accuracy by optimizing LSTM model parameters. Mehtab and Sen (2020) used CNN and LSTM-based deep learning models for stock price prediction. Li et al. (2020) proposed a stock forecasting model FS-LSTM based on the 5G Internet of Things, combining feature selection and LSTM. Bose et al. (2021) developed a hybrid model by cascading Multivariate Adaptive Regression Splines (MARS) and Deep Neural Network (DNN) to predict closing prices of stock. Chen et al. (2024) introduced a multi-feature stock price prediction model based on multi-feature calculation, LASSO feature selection, and the Ca-LSTM network. Gao et al. (2021) compared LSTM and GRU models for stock price prediction under different dimension reduction methods. Joel et al. (2023) proposed an Island Parallel-Harris Hawks Optimizer (IP–HHO)-optimized Convolutional LSTM (ConvLSTM) with an autocorrelation model. Li et al. (2023) developed a novel LASSO-ATT-LSTM model for stock price prediction based on multi-source heterogeneous data. Liao et al. (2024) introduced LEET, a stock market forecast model with long-term emotional change-enhanced temporal analysis. Zhang et al. (2024) proposed SMPDF, a stock movement prediction model based on stock prices and text data. Kumar Chandar and Punjabi (2022) created an effective stock market prediction model combining technical indicators from historical data and an Elman neural network (ENN). Luo and Ji (2022) developed a hybrid stock price prediction model using improved particle swarm optimization (IPSO) and LSTM.

2.4. Novel approaches and recent trends

Recent research has explored novel approaches to stock price prediction, incorporating diverse data sources and advanced techniques.

Zhang et al. (2024) introduced SMPDF, a stock movement prediction model based on stock prices and text data, highlighting the growing trend of incorporating textual information. Zhu et al. (2024)presented PMANet, an advanced hybrid model for stock price prediction based on Multi-scale Timing Feature Attention. Li et al. (2024) proposed a DeepAR model based on the attention mechanism (DeepARA) for both single-point and probabilistic predictions of stock prices. Liao et al. (2024) developed LEET, a stock market forecast model with long-term emotional change-enhanced temporal analysis, incorporating sentiment analysis into prediction models. Chen et al. (2024) introduced a multi-feature stock price prediction model based on multi-feature calculation, LASSO feature selection, and the Ca-LSTM network. Li et al. (2023) developed a novel LASSO-ATT-LSTM model for stock price prediction based on multi-source heterogeneous data. Mian (2023) evaluated stock closing prices using Transformer learning, comparing it with traditional and deep learning methods. Jafari and Haratizadeh (2022) introduced GCNET, a graph-based prediction model for stock price movement using graph convolutional networks. Staffini (2022) introduced a Deep Convolutional Generative Adversarial Network (DCGAN) architecture for forecasting the closing price of stocks. Chandar (2021) developed a hybrid model to improve stock price prediction utilizing ANFIS variations. DeepAR, introduced as a probabilistic forecasting methodology, employs autoregressive recurrent neural networks trained on multiple related time series. The model's capability to produce accurate probabilistic forecasts has been demonstrated particularly in retail business applications, where precise demand forecasting is crucial for inventory optimization. DeepAR's architecture overcomes traditional forecasting challenges through its deep learning approach, requiring minimal manual intervention while maintaining superior accuracy (J. Li et al., 2024; Salinas et al., 2020).

The N-HITS model addresses long-horizon forecasting challenges through innovative hierarchical interpolation and multi-rate data sampling techniques. The model's architecture enables sequential prediction assembly, emphasizing components with varying frequencies and scales. N-HITS has demonstrated remarkable efficiency, achieving an average accuracy improvement of 20% compared to transformer architectures while reducing computational complexity by a factor of 50. The model's theoretical foundation is supported by proven approximation capabilities for extended forecast horizons under smoothness conditions (Challu et al., 2023).

N-BEATS represents a significant advancement in univariate time series forecasting through its deep neural architecture, incorporating backward and forward residual links with fully-connected layers. This architecture has demonstrated exceptional versatility across diverse domains, improving forecast accuracy by 11% over statistical benchmarks. The model's success without domain-specific components challenges conventional wisdom, suggesting that deep learning primitives alone can effectively address various forecasting challenges (Oreshkin et al., 2020).

Recent developments in gradient-boosted trees, particularly highlighted in M5 competition solutions, have established new benchmarks in forecasting accuracy. These approaches address critical challenges in time series forecasting, including overfitting, leakage, and non-stationarity. The integration of cross-validation strategies, augmentation techniques, and parameter tuning has proven highly effective, leading to winning solutions in major forecasting competitions. Hybrid approaches combining gradient-boosted trees with neural networks have shown particular promise, especially in hierarchical time series forecasting. These hybrid models demonstrate superior performance in both point and probabilistic forecasting tasks, emphasizing the importance of diverse model ensembles and careful validation set construction (Chiew & Choong, 2022; Lainder & Wolfinger, 2022; H. Li et al., 2020).

This literature review section demonstrates the diverse approaches and ongoing advancements in stock price prediction, ranging from traditional time series methods to cutting-edge deep learning and hybrid models. The field continues to evolve, with researchers exploring novel techniques to improve prediction accuracy and incorporate diverse data sources. The review of the literature on stock price prediction reveals several key trends and gaps in the field. Traditional time series methods, such as ARIMA, continue to be relevant but are increasingly being complemented or replaced by more advanced techniques. Machine learning approaches, particularly ensemble methods and support vector machines, have shown promising results in capturing non-linear relationships in financial data.

Deep learning models, especially those based on LSTM architectures, have emerged as powerful tools for stock price prediction due to their ability to capture long-term dependencies in time series data. The trend towards hybrid and ensemble models, combining multiple techniques, reflects the complex nature of stock price movements and the need for diverse approaches to capture different aspects of market behavior.

Recent research has also focused on incorporating diverse data sources, including textual data from news and social media, to provide a more comprehensive view of factors influencing stock prices. Additionally, there's an increasing emphasis on developing models that can provide not just point predictions, but also uncertainty estimates and probabilistic forecasts.

Despite these advancements, several gaps remain in the literature:

1. While many studies have applied various optimization techniques to LSTM models, the potential of the SCSO algorithm in this context has not been explored.
2. The application of 2D and 3D LSTM architectures to stock price prediction, despite their success in other domains, has been limited.
3. There's a need for more comprehensive comparisons of different deep learning architectures and optimization techniques on a common dataset to establish clear benchmarks.
4. The economic value and practical applicability of advanced prediction models in real-world trading scenarios are often not thoroughly addressed.

The present study aims to address these gaps by introducing a novel LSTM-SCSO model, comparing it with other LSTM variants including 2D and 3D architectures, and providing a thorough analysis of its performance and potential economic implications. This approach not only contributes to methodological advancement in the field but also aims to bridge the gap between academic research and practical application in stock price prediction.

3. Material and method

3.1. Time series analysis

In time series analysis, data collected over time is explored and analyzed to understand and identify patterns, trends, and seasonality. Trend refers to the long-term upward or downward movement in a time series. It shows the overall direction in which the data is moving over an extended period. It helps to understand the underlying growth or decline in the variable being studied (Bi et al., 2021).

Seasonality, on the other hand, refers to a repeating pattern that occurs within a specific time frame. It can be a daily, weekly, monthly, or yearly pattern. Seasonal components often occur due to various factors, such as weather, holidays, or events that impact the data regularly and predictably (Kim & Kim, 2021).

To analyze time series data, often used statistical techniques such as decomposition, smoothing, and forecasting models. Decomposition helps in separating the time series into its components, including trend, seasonality, and residuals (random variations). This enables to examination and analysis of each component separately, gaining insights into their individual contributions to the overall pattern (Talavera-Llames et al., 2018).

Smoothing techniques, like moving averages and exponential smoothing, are used to reduce noise and highlight the underlying patterns in the data. They provide a clearer view of the trend and seasonality by removing random fluctuations (Hamilton, 2020).

Overall, time series analysis, with its components of trend, seasonality, and forecasting, allows one to gain valuable insights, make informed decisions, and predict future behavior based on the patterns observed in the data (Almasarweh & Alwadi, 2018).

3.2. ANN

Artificial neural networks (ANNs) are a class of machine learning models inspired by the structure and functioning of biological neural networks. They are designed to mimic the behavior of interconnected neurons in the human brain to perform complex computational tasks. ANNs consist of multiple layers of artificial neurons, known as nodes or units, organized into an input layer, one or more hidden layers, and an output layer (Gülcü, 2022).

The process of using ANNs involves training the network on a labeled dataset to learn the underlying patterns and relationships within the data. During training, the weights and biases of the connections between the nodes are adjusted iteratively through a process called backpropagation. Backpropagation calculates the gradients of the network's error with respect to its weights, allowing the model to update and refine its parameters to minimize prediction errors.

The nodes in an ANN receive inputs, apply activation functions, and produce outputs that are passed to the next layer. The activation functions introduce non-linearities to enable the network to capture complex relationships between inputs and outputs. Standard activation functions include sigmoid, tanh, and ReLU (Rectified Linear Unit).

The power of ANNs lies in their ability to learn from complex and high-dimensional data, extract intricate patterns, and make accurate predictions or classifications. They have been successfully applied to a wide range of tasks, including image recognition, natural language processing, speech recognition, and stock market prediction.

However, it's important to note that ANNs require careful architectural design, parameter tuning, and substantial amounts of labeled training data to achieve optimal performance. Overfitting, a phenomenon where the model memorizes the training data without generalizing well to unseen data, is also a concern. Regularization techniques and validation strategies are used to address overfitting and ensure the model's generalization capability (Dengiz et al., 2009).

3.3. LSTM

LSTM (Long Short-Term Memory) is a type of RNN architecture that is specifically designed to handle long-term dependencies in sequential data. It addresses the vanishing gradient problem in traditional RNNs and allows for more effective learning and prediction (Mehtab & Sen, 2020).

LSTM units have a more complex structure compared to standard neurons in RNNs. They consist of a cell state, three gating mechanisms (input gate, forget gate, and output gate), and an activation function. These components work together to regulate the flow of information through the network. The forget gate determines how much of the previous cell state is retained or forgotten. The input gate determines which parts of the current input are essential to update the cell state. The candidate cell state computes the new candidate values to be added to the cell state. The cell state is updated by combining the forget gate, input gate, and candidate cell state. The output gate determines how much of the updated cell state is exposed as the output. Finally, the output of the LSTM unit is computed by multiplying the updated cell state with the output gate after applying an activation function, typically a hyperbolic tangent or a sigmoid function (Yuan et al., 2023).

LSTM effectively captures and retains long-term dependencies in sequential data, making it a powerful tool for tasks such as language modeling, speech recognition, and time series prediction (Mittal, Kumar, Roy, Balasubramanian, & Chaudhuri, 2019).

3.4. Sand Cat swarm algorithm

The Sand Cat Swarm Optimization (SCSO) algorithm is a nature-inspired metaheuristic optimization technique based on the hunting behavior of sand cats (Seyyedabbasi & Kiani, 2023). In the context of stock price prediction, SCSO is used to optimize the parameters of the LSTM model, enhancing its predictive capabilities.

Key features of SCSO relevant to LSTM optimization include:

1. Adaptive search range: SCSO employs a gradually decreasing sensitivity range, allowing for broad exploration in early iterations and fine-tuning in later stages. This feature helps in finding optimal LSTM parameters by balancing exploration and exploitation.
2. Position update mechanism: The algorithm updates the position of search agents (representing LSTM parameters) based on the best-known solution and current position, enabling efficient navigation of the parameter space.
3. Randomized angle selection: SCSO uses a random angle to determine the direction of movement, helping to avoid local optima and explore diverse regions of the parameter space.

In the context of LSTM for stock price prediction, SCSO addresses several challenges:

1. Hyperparameter tuning: SCSO optimizes LSTM hyperparameters such as the number of hidden layers, neurons per layer, and learning rate, which are crucial for model performance.
2. Feature selection: The algorithm can be used to identify the most relevant input features for stock price prediction, improving model efficiency and accuracy.
3. Overfitting prevention: By optimizing the model's architecture and regularization parameters, SCSO helps in building LSTM models that generalize well to unseen data.

The SCSO-optimized LSTM model aims to capture complex temporal dependencies in stock price data more effectively than traditional optimization methods, potentially leading to improved prediction accuracy and robustness.

3.5. Models

3.5.1. ANN

Table 1 presents the architecture of the ANN model utilized in this study. The network consists of four layers, starting with an input layer with 20 nodes. Following the input layer are two dense layers, each comprising 10 nodes, enabling the model to learn complex patterns and representations from the data. A dropout layer with a dropout rate of 0.5 is incorporated after the second dense layer to prevent overfitting and

Table 1
ANN model architecture.

Layer	Parameter
Input	20
Dense	10
Dense	10
Dropout	0.5
Dense	1

enhance generalization. Finally, the output layer consists of a single node appropriate for the specific task at hand. This ANN architecture is carefully designed to balance complexity and simplicity, aiming to achieve accurate predictions while avoiding overfitting issues.

3.5.2. LSTM1D

Table 2 illustrates the architecture of the LSTM1D model used in this study. The model is well-suited for sequence data because it captures long-term dependencies. The input layer contains 20 nodes, allowing the model to process sequences of a certain length. Next, an LSTM layer with 10 memory cells is employed to extract relevant temporal patterns from the data. Following the LSTM layer, there is a dense layer with 10 nodes, facilitating the learning of complex representations. To prevent overfitting, a dropout layer with a dropout rate of 0.5 is included, which aids in enhancing the generalization of the model. Finally, the output layer consists of a single node suitable for the specific task under consideration. The LSTM1D architecture is thoughtfully designed to handle sequential data effectively, enabling accurate predictions and capturing the underlying temporal dynamics of the dataset.

3.5.3. LSTM2D

Table 3 presents the architecture of the LSTM2D model used in this study. The LSTM2D model is specifically designed to handle two-dimensional sequential data, such as time series data with multiple features or images with temporal dependencies. The input layer consists of 20 nodes, which allows the model to process sequences with a certain length and multiple features. Subsequently, two LSTM layers are employed, each with 10 memory cells. These LSTM layers enable the model to effectively capture complex temporal patterns and dependencies in the data. Following the LSTM layers, a dense layer with 10 nodes is utilized to learn higher-level representations from the extracted features. A dropout layer with a dropout rate of 0.5 is incorporated to prevent overfitting and improve generalization. Finally, the output layer consists of a single node well-suited for the specific task at hand. The LSTM2D architecture is carefully crafted to handle the complexities of two-dimensional sequential data, aiming to achieve accurate predictions while capturing the underlying temporal relationships in the dataset.

3.5.4. LSTM3D

Table 4 presents the architecture of the LSTM3D model used in this study. The LSTM3D model is specifically designed to handle three-dimensional sequential data, such as spatiotemporal data or videos with temporal dependencies and spatial structures. The input layer contains 20 nodes, allowing the model to process sequences with a certain length, multiple features, and spatial dimensions. Subsequently, three LSTM layers are utilized, each with 10 memory cells. These LSTM layers enable the model to effectively capture complex spatiotemporal patterns and dependencies. The LSTM3D architecture is particularly well-suited for tasks where the data's temporal and spatial aspects are critical for accurate predictions. After the LSTM layers, a dense layer with 10 nodes is employed to learn higher-level representations from the extracted spatiotemporal features. A dropout layer with a dropout rate of 0.5 is included to prevent overfitting and enhance generalization. Finally, the output layer consists of a single node suitable for the specific task under consideration. The LSTM3D architecture is thoughtfully designed to handle the intricacies of three-dimensional sequential data, aiming to achieve accurate predictions while capturing the underlying spatiotemporal relationships in the dataset.

Table 2
LSTM1D architecture.

Layer	Parameter
Input	20
LSTM	10
Dense	10
Dropout	0.5
Dense	1

Table 3
LSTM2D architecture.

Layer	Parameter
Input	20
LSTM	10
LSTM	10
Dense	10
Dropout	0.5
Dense	1

Table 4
LSTM3D architecture.

Layer	Parameter
Input	20
LSTM	10
LSTM	10
LSTM	10
Dense	10
Dropout	0.5
Dense	1

3.5.5. Auto-ARIMA

Autoregressive Integrated Moving Average (ARIMA) models are implemented as statistical benchmarks for time series forecasting. The Auto-ARIMA approach automatically identifies the optimal parameters for the ARIMA (p,d,q) model, where p represents the order of the autoregressive term, d is the degree of differencing, and q is the order of the moving average term. This automated process eliminates subjective parameter selection and ensures reproducibility.

The Auto-ARIMA algorithm employs a stepwise approach to determine the best combination of parameters. The process begins with the evaluation of the time series' stationarity through unit root tests. The degree of differencing (d) is determined based on these tests. Subsequently, the algorithm performs a grid search over different combinations of p and q values. The Akaike Information Criterion (AIC) is utilized as the primary metric for model selection, as it provides a balanced assessment of model complexity and goodness of fit.

The implementation includes seasonal components when necessary, extending the model to SARIMA (p,d,q) (P, D, Q)m form, where P, D, and Q represent the seasonal equivalents of p, d, and q, and m denotes the seasonal period. The algorithm also incorporates drift terms when appropriate, based on trend analysis of the time series (Almasarweh & Alwadi, 2018).

3.5.6. Gradient Boosted Trees (GBT)

Gradient Boosted Trees, implemented through the XGBoost framework, serve as a machine learning benchmark in this study. GBT operates by constructing an ensemble of decision trees sequentially, where each subsequent tree aims to correct the prediction errors of its predecessors. This approach is particularly effective for time series forecasting due to its ability to capture non-linear relationships and handle multiple input features simultaneously. The implementation utilizes a gradient-boosting framework that optimizes a regularized objective function. This function combines a convex loss function with a regularization term to control model complexity. The model processes the technical indicators and macroeconomic variables as features, maintaining their temporal order in the training process. Early stopping is implemented to prevent overfitting, where the model training terminates if no improvement is observed in the validation set for a specified number of rounds. Feature importance analysis is incorporated into the GBT framework, providing insights into the relative significance of different technical and macroeconomic indicators. The model employs a tree-specific learning rate to control the contribution of each tree to the final prediction. Cross-validation is performed using a time-series-aware splitting strategy to maintain the temporal structure of the data (Chiew & Choong, 2022; Lainder & Wolfinger, 2022; Nasios & Vogklis, 2022).

3.5.7. DeepAR

DeepAR, developed by Amazon Research, represents a probabilistic forecasting model based on autoregressive recurrent networks. The model combines deep learning with statistical forecasting principles to generate probabilistic predictions. This architecture is particularly suitable for financial time series due to its ability to capture complex patterns and provide uncertainty estimates in its forecasts.

The implementation consists of multiple LSTM layers that process the input sequences of historical values and associated features. DeepAR learns a global model across time series in the dataset, enabling it to learn patterns that are shared across different stocks. The model outputs parameters of a probability distribution, typically Gaussian, for each time step in the forecast horizon. This probabilistic approach provides not only point forecasts but also confidence intervals for the predictions.

The training process employs a maximum likelihood estimation approach, where the model learns to maximize the probability of the observed data given the predicted distribution parameters. The architecture incorporates time-varying features, including the technical indicators and macroeconomic variables, as additional inputs to the network. Temporal dependencies are maintained through the sequential processing of input data, while the model's probabilistic nature allows for the quantification of prediction uncertainty (Salinas et al., 2020).

3.5.8. N-BEATS

Neural Basis Expansion Analysis for Time Series (N-BEATS) represents a deep neural architecture specifically designed for time series forecasting. The model employs a unique architectural design based on backward and forward residual links, with no direct use of traditional time series decomposition techniques. This deep learning approach processes raw time series data through specialized blocks that automatically learn decomposition patterns.

The architecture consists of multiple stacks of basic blocks, where each block contains fully connected layers that terminate in two branches: the backcast and forecast branches. The backcast branch reconstructs the input sequence, while the forecast branch generates future predictions. This dual-output mechanism enables the model to learn hierarchical pattern representations. Each subsequent block processes the residual error from the previous block's backcast, creating an iterative refinement process.

The implementation utilizes double residual stacking, where each stack specializes in different aspects of the time series. The first stack typically captures trend components, while subsequent stacks learn seasonal and higher-frequency patterns. The model processes the technical indicators and macroeconomic variables as additional input features, maintaining their temporal relationships. The architecture's design allows for interpretable outputs, as each block's contribution to the final forecast can be analyzed separately (Oreshkin et al., 2020).

3.5.9. N-HITS

Neural Hierarchical Interpolation for Time Series (N-HITS) represents an enhancement of the N-BEATS architecture, specifically designed to incorporate interpretable time series components. The model introduces a hierarchical interpolation structure that explicitly models trend, seasonal, and irregular components of time series data. This architecture is particularly effective for financial time series due to its ability to capture multiple temporal resolutions and provide interpretable decompositions.

The implementation employs a specialized stack architecture where each stack is dedicated to modeling specific time series components. The trend stack utilizes polynomial interpolation to capture long-term patterns, while the seasonal stack implements periodic interpolation for cyclical patterns. The model incorporates additional stacks for capturing irregular components and short-term fluctuations. This hierarchical approach enables the model to process both low-frequency patterns (such as macroeconomic trends) and high-frequency components (such as daily price movements) simultaneously.

N-HITS extends the traditional N-BEATS framework by incorporating interpretable basis functions and maintaining explicit separation between different temporal resolutions. The model processes the technical indicators and macroeconomic variables through specialized input layers, preserving their temporal relationships. The architecture's interpretability is enhanced through component-wise analysis, where the contribution of each stack to the final prediction can be visualized and analyzed separately (Challu et al., 2023).

3.5.10. LSTM-GA and LSTM-SCSO

Table 5 illustrates the architectures of two different LSTM-based models: LSTM-GA and LSTM-SCSO. The LSTM-GA and LSTM-SCSO models are designed for a specific task where the exact number of LSTM layers and the number of nodes in each layer are determined by GA and SCSO. The model begins with an input layer consisting of 20 nodes, allowing it to process sequences of a certain length and multiple features. Instead of specifying the number of LSTM layers and nodes directly, they are represented by variables x0, x1, x2, x3, x4, x5, and x6, which will be determined by the algorithm during the model optimization process. The LSTM-GA and LSTM-SCSO architectures allow the metaheuristic algorithms to discover the optimal number of LSTM layers and nodes, providing a flexible and adaptive model for various data and tasks.

3.6. Justification for LSTM2D and LSTM3D usage

While LSTM2D and LSTM3D are typically associated with processing two-dimensional and three-dimensional data respectively, their application in stock price prediction offers several advantages:

1. Multi-feature analysis: LSTM2D can process multiple features simultaneously, treating each feature as a separate dimension. This allows for a more comprehensive analysis of stock price movements concerning various indicators.
2. Temporal-spatial correlations: LSTM3D can capture both temporal and spatial correlations in the data. In stock price prediction, this can be useful for analyzing relationships between different stocks or market sectors over time.
3. Enhanced pattern recognition: The additional dimensions in LSTM2D and LSTM3D enable the models to recognize more complex patterns that may not be apparent in one-dimensional time series data.
4. Improved feature extraction: These models can automatically extract hierarchical features from the input data, potentially uncovering hidden relationships in stock price movements.
5. Handling of multi-scale temporal dynamics: LSTM2D and LSTM3D can capture both short-term and long-term dependencies in the data, which is crucial for stock price prediction where different time scales may influence price movements.

Table 5
LSTM-GA and LSTM-SCSO architectures.

Layer	Parameter
Input	20
LSTM	x_0
LSTM (x_1)	x_2
LSTM (x_3)	x_4
Dense	x_5
Dropout	x_6
Dense	1

While the input data for stock price prediction is primarily time-series, it is reshaped to fit the 2D and 3D input requirements of these models. This reshaping allows us to leverage the additional capabilities of LSTM2D and LSTM3D in capturing complex patterns and relationships in the data.

3.7. Hyperparameter alternatives

Many existing studies employ machine learning methods with manual tuning and fixed parameters. This approach has significant disadvantages. Manual tuning requires researchers to have deep domain knowledge, understand dynamic market changes, and adjust parameters accordingly. This situation is difficult for beginners and non-experts, which limits the widespread use of the method. Manual tuning may lead to human bias, as researchers' subjective decisions affect parameter selection, and the objectivity and universality of the model are affected. Hyperparameter optimization increases the success of models. Therefore, hyperparameter optimization has been applied in this study. This study presents a more flexible tuning strategy to increase the model's adaptability and robustness (Feng et al., 2024).

Table 6 displays the alternative hyperparameters and their respective options considered for model optimization:

Number of neurons: This hyperparameter governs the number of neurons (nodes) in the dense layers of the model. The alternatives range from 1 to 20, providing flexibility in choosing the size of the dense layers.

Layer exists or does not exist: This binary hyperparameter determines whether an additional dense layer is present in the model or not. The alternatives are represented by 0 (not exist) and 1 (exist), allowing for an investigation into the necessity of including extra dense layers.

Dropout rate: Dropout is a regularization technique that randomly drops out some neurons during training to prevent overfitting. The alternatives for the dropout rate are 0.3, 0.4, 0.5, 0.6, and 0.7, offering various levels of dropout intensity.

Optimizer algorithm: The optimizer is responsible for updating the model parameters during training to minimize the loss function. The alternatives for the optimizer algorithm include Adagrad, Adam, Adamax, RMSprop, and SGD, providing a range of optimization techniques to choose from.

Learning rate: The learning rate determines the step size at which the optimizer adjusts the model parameters. It is a critical hyperparameter that affects the training process. The alternatives for the learning rate are 0.01, 0.001, 0.0001, 0.00001, and 0.000001, covering a broad spectrum of learning rates to find the most suitable value for efficient convergence.

Considering these alternative hyperparameters, model selection is performed using GA and SCSO to find the best combination of hyperparameters that maximizes the model's performance on the validation data.

3.8. Input data and feature extraction

The study incorporates both historical stock prices and key technical and macroeconomic indicators as input data for the predictive models. The technical indicators are calculated using historical price data to capture market trends and momentum. These include Simple Moving Averages (5-day, 10-day, 20-day, and 50-day) to identify trend directions, Relative Strength Index (14-day RSI) to measure momentum, Moving Average Convergence Divergence (12-day and 26-day moving averages with 9-day signal line) to identify trend changes, Volume Moving Average (20-day) to analyze trading volume patterns, and Rate of Change (10-day) to measure price momentum.

Table 6
Hyperparameter alternatives.

Hyperparameter	Alternatives
Number of neurons	1, 2, 3, …, 18, 19, 20
Layer exist or not exist	0, 1
Dropout rate	0.3, 0.4, 0.5, 0.6, 0.7
Optimizer algorithm	Adagrad, Adam, Adamax, RMSprop, SGD
Learning rate	0.01, 0.001, 0.0001, 0.00001, 0.000001

Several factors may enhance predicting accuracy individually, although their impact is limited. Augmenting the number of predictors from designated categories improves forecasting precision (Fu et al., 2024). Therefore, macroeconomic indicators from Deutsche Bundesbank are integrated to capture broader market conditions. These include German short-term interest rates (3-month rates), German inflation rates (Consumer Price Index), and German unemployment rates. These monthly indicators provide context for the overall economic environment affecting stock prices.

The LSTM architecture effectively processes this diverse set of inputs due to its ability to capture long-term dependencies across multiple features. The model learns complex relationships between technical indicators, macroeconomic factors, and stock prices, identifying patterns that may not be apparent through traditional analysis. The architecture's capacity to handle non-linear relationships is particularly valuable given the complex interactions between different economic indicators and stock price movements.

The combination of technical and macroeconomic indicators with historical price data provides a more comprehensive view of market conditions. This multi-feature approach allows the model to consider both market-specific technical factors and broader economic conditions in making predictions (Feng et al., 2024; Fu et al., 2024). The LSTM's adaptive feature extraction capabilities enable it to determine the relative importance of different indicators under varying market conditions, potentially uncovering hidden relationships between these diverse inputs. This enhanced input framework represents an improvement over approaches that rely solely on historical prices, as it incorporates established predictive factors from both technical and fundamental analysis. The integration of these additional features provides the model with a richer context for price prediction while maintaining the LSTM's ability to automatically extract and learn from complex patterns in the data.

4. Results and discussion

4.1. Dataset

The DAX, also known as the Deutscher Aktienindex, has a notable historical background that commenced in 1988 with its introduction by the Frankfurt Stock Exchange. Its primary purpose is to assess the performance of Germany's most prominent and highly liquid corporations. Over time, the DAX has emerged as an essential gauge for the German economy and a prominent indicator of European stock market trends. Notable achievements have punctuated the trajectory of its progression. Throughout its existence, the DAX has experienced a range of market fluctuations. Presently, the DAX remains a prominent participant in the realm of finance, garnering international recognition and exerting an impact on investor perception. The DAX, functioning as a price-weighted index, reflects the performance exhibited by its constituent companies. These companies are representative of diverse sectors within the German economy. The real-time performance of this indicator is closely monitored by investors, financial institutions, and policymakers due to its significance as a crucial economic indicator within a dynamic financial environment (Banke et al., 2022; Bühler & Kempf, 1995; Henne et al., 2009; Stapf & Werner, 2003). Historical prices of 30 stocks in the DAX index are obtained using Yahoo Finance. A total of 5 years of data is used. This data covers the period between 2018 and 2023. Forecasts are made based on the closing price for each stock.

The evaluation framework employs a rigorous rolling window approach with time-series cross-validation to ensure robust model assessment. A 20-day rolling window is utilized, where models are trained on a moving window of historical data and tested on subsequent periods. This approach better reflects real-world trading conditions and prevents look-ahead bias. The dataset is structured with an 80-20 split, where the final 20% is reserved for out-of-sample testing.

Time-series cross-validation is implemented through a forward-chaining methodology. The initial training period consists of the first 80% of the data, with subsequent validation performed on the next available period. This process is repeated by moving the window forward, maintaining the temporal order of the data, and ensuring the validity of the forecasting framework.

The models' performance is evaluated using out-of-sample R2 scores, which provide a more realistic assessment of predictive capability compared to in-sample measures. This metric is calculated exclusively on the test set data, offering a true measure of the models' generalization ability. The rolling window approach, combined with time-series cross-validation and out-of-sample evaluation, provides a comprehensive framework for assessing the models' practical forecasting capabilities.

4.2. Evaluation criteria

Regression evaluation criteria are employed to evaluate the efficacy of regression models, which are utilized to forecast continuous numerical outcomes. The following is a concise elucidation of each evaluation metric:

MSE is a metric used to evaluate the accuracy of a predictive model. It quantifies the average of the squared differences between the predicted values and the actual values within a given dataset. The weighting of larger errors is more severe, resulting in a heightened sensitivity to outliers. Smaller MSE values are indicative of superior model performance, with a value of 0 representing the optimal outcome where predictions precisely align with the actual values.

The coefficient of determination, often denoted as out of sample R2, is a statistical measure used in regression analysis to assess how well a regression model fits the observed data. It provides a quantifiable indication of the proportion of the variability in the dependent variable that can be explained by the independent variables included in the model.

MAE is a metric used to measure the average absolute difference between predicted and actual values. The method under consideration exhibits reduced sensitivity to outliers in comparison to the MSE approach, as it does not involve squaring the errors. Similar to the RMSE and MSE, lower values of MAE are indicative of superior model performance.

MAPE is a metric used to determine the average percentage deviation between predicted and actual values. The statement as mentioned above denotes the measure of the discrepancy concerning the true values. MAPE is frequently employed as a metric to quantify the level of accuracy in predictions, typically represented as a percentage value. Similar to other evaluation metrics, lower MAPE values are indicative of superior model performance.

In the assessment of regression models, it is imperative to take into account various metrics in order to obtain a comprehensive comprehension of their performance. The selection of the most suitable metric is contingent upon the particular context and demands of the problem at hand.

MSE, MAE, MAPE, out of sample R2 as evaluation criteria for this stock price prediction model is based on the following considerations:

1. Comprehensive error assessment: MSE and MAE provide complementary insights into prediction errors. MSE is sensitive to large errors, making it suitable for detecting significant deviations, while MAE offers a more intuitive measure of average error magnitude.
2. Relative performance measure: MAPE allows for comparison across different stocks by expressing errors as percentages, facilitating the interpretation of model performance across various price scales.
3. Explanatory power: out of sample R2 score indicates how well the model explains the variance in stock prices, offering insight into the model's overall predictive capability.
4. Alignment with financial domain: These metrics are widely used and understood in financial forecasting, making the results comparable with existing literature and industry standards.
5. Data characteristics: Given the continuous nature of stock price data and its potential for outliers, this combination of metrics provides a balanced view of model performance, addressing both absolute and relative errors.

By using these diverse metrics, it is aimed to provide a comprehensive evaluation of the models' performance, catering to different aspects of prediction accuracy relevant to stock market analysis.

4.3. Result and discussion

This study examines the performance of several algorithms, namely LSTM1D, LSTM2D, LSTM3D, ANN, LSTM-GA, Auto-ARIMA, GBT, DeepAR, N-BEATS, N-HITS, and optimized LSTM with SCSO (LSTM-SCSO), when applied to the DAX stock dataset spanning the period from 2018 to 2023. Subsequently, the outcomes of these algorithms are meticulously examined and documented. The objective of this study is to ascertain the optimal algorithm in terms of both accuracy and efficiency for the prediction of stock prices. The evaluation criteria employed encompass MSE, MAE, MAPE, and out-of-sample R2 scores. The outcomes of the testing hold significant importance in the DAX market.

Upon analyzing Table 7, it is evident that the LSTM-SCSO algorithm consistently achieves the lowest MSE values across most of the stock tickers. This indicates that the LSTM-SCSO model outperforms the other algorithms in terms of MSE. LSTM-SCSO's superior performance may be attributed to its ability to capture complex temporal dependencies and patterns in the data, making it more effective in predicting the continuous numerical outcomes of the stock tickers; however, while LSTM-SCSO appears to be the best algorithm based on the MSE values in this stock price dataset. LSTM-GA follows it, and it is the second-best algorithm.

Table 8 focuses on MAE comparison. Similar trends emerge with LSTM-SCSO consistently outperforming other methods for many stocks. It's worth noting that while LSTM-based algorithms generally exhibit better predictive capabilities, certain stocks showcase varying algorithmic preferences, emphasizing the need for tailored algorithm selection. These tables collectively provide valuable insight into algorithmic performance across diverse stocks, aiding in informed decision-making for stock market prediction applications. LSTM-SCSO model performs the best result. It is the best algorithm for the stock market price prediction dataset for MAE criteria. LSTM-GA follows it.

Table 9 adds to the comprehensive analysis of algorithm performance by presenting MAPE comparison for the evaluated algorithms, including LSTM-SCSO, LSTM-GA, LSTM1, LSTM2, LSTM3, ANN, Auto-ARIMA, Gradient Boosted Trees, DeepAR, N-BEATS, N-HITS. The MAPE metric provides insights into the relative accuracy of the algorithms' predictions, accounting for the percentage difference between predicted and actual values. Notably, LSTM-SCSO consistently exhibits the lowest MAPE values across various stock tickers, indicating their superior predictive accuracy in comparison to other algorithms. These findings reinforce the dominance of LSTM-SCSO in terms of prediction precision, underlining their suitability for stock market forecasting tasks. However, it's important to acknowledge that each algorithm's performance varies depending on the specific stock being analyzed, reinforcing the importance of tailored algorithm selection for optimal predictive outcomes. LSTM-SCSO model performs the best result. It is the best algorithm for the stock market price prediction dataset for

Table 7
MSE comparison of the algorithms.

MSE											
Ticker	LSTM-SCO	LSTM-GA	LSTM-1D	LSTM-2D	LSTM-3D	ANN	ARIMA	GBT	DeepAR	N-BEATS	N-HITS
ALV.DE	28.153	33.940	40.889	75.945	60.564	180.404	181.139	170.487	57.809	61.349	102.957
HNR1.DE	11.496	25.027	58.304	68.592	64.213	79.311	81.664	109.320	33.953	67.176	54.859
P911.DE	17.138	24.181	77.915	68.731	65.140	73.136	41.453	82.536	80.978	80.944	48.528
FRE.DE	3.584	4.418	7.708	13.028	25.140	9.209	17.558	5.532	10.599	11.342	7.379
CON.DE	15.469	43.162	126.997	91.947	150.070	50.689	94.186	127.112	90.503	69.287	37.267
IFX.DE	0.998	1.573	4.224	3.815	4.073	6.818	10.139	4.479	2.305	4.284	2.174
DTE.DE	0.152	0.413	1.252	0.701	1.335	0.881	1.180	1.228	1.551	0.679	1.136
SIE.DE	17.474	21.591	54.184	60.717	53.725	90.505	62.933	36.446	46.254	52.248	33.548
DBK.DE	0.148	0.164	0.737	0.743	0.476	0.658	0.684	0.720	0.720	0.579	0.519
AIR.DE	11.944	11.816	34.946	29.694	35.986	42.398	41.804	52.111	25.019	33.955	21.700
HEI.DE	1.559	3.445	8.935	6.209	6.031	8.546	9.665	7.246	8.410	5.227	6.897
BAYN.DE	1.706	3.002	5.211	11.174	8.270	2.431	11.044	4.437	3.539	3.843	6.856
EOAN.DE	0.025	0.039	0.137	0.158	0.127	0.142	0.111	0.109	0.139	0.091	0.107
BMW.DE	5.664	7.497	23.560	37.369	12.516	64.019	84.630	20.092	20.961	32.458	8.119
RWE.DE	3.327	1.601	7.976	5.687	16.070	22.713	18.153	6.399	8.693	11.066	9.409
VOW3.DE	10.740	11.111	34.077	46.225	36.208	25.296	34.324	20.767	24.393	26.543	16.842
DTG.DE	0.556	1.188	3.612	5.101	5.698	3.988	6.299	3.059	2.942	2.670	3.506
VNA.DE	0.965	1.721	2.747	4.205	5.898	4.981	4.841	5.079	3.192	4.369	2.371
ENR.DE	2.592	2.171	4.952	9.269	16.868	3.052	6.448	5.058	4.801	3.371	3.337
SY1.DE	7.134	15.008	23.775	15.613	17.137	11.299	33.594	20.320	24.888	16.347	20.661
BEI.DE	1.698	2.266	9.245	4.422	8.551	12.230	15.724	5.143	5.133	8.270	10.567
ADS.DE	105.623	96.036	351.687	507.183	685.381	422.921	595.587	304.360	240.560	238.348	345.142
SHL.DE	1.487	2.073	3.777	3.951	5.093	19.002	6.888	6.383	3.966	6.003	4.066
ZAL.DE	4.159	6.783	33.780	27.177	27.307	64.283	30.625	31.258	18.031	14.090	20.660
DB1.DE	14.237	48.718	87.780	52.749	150.839	106.501	46.350	123.319	155.219	82.985	79.301
1COV.DE	1.214	1.642	4.725	5.629	4.270	43.520	8.312	5.768	4.497	7.456	3.256
BAS.DE	2.051	1.822	4.843	4.727	4.867	6.442	8.468	2.967	4.458	3.765	5.534
MTX.DE	21.331	26.126	82.327	74.533	132.341	126.326	87.832	92.936	97.357	144.997	54.152
MRK.DE	14.268	23.393	90.441	44.782	43.986	65.434	84.772	50.284	80.262	55.058	78.115

Table 8
MAE comparison of the algorithms.

MAE											
Ticker	LSTM-SCO	LSTM-GA	LSTM-1D	LSTM-2D	LSTM-3D	ANN	ARIMA	GBT	DeepAR	N-BEATS	N-HITS
ALV.DE	4.377	4.768	5.170	7.140	6.504	11.970	11.245	11.790	6.264	6.544	8.322
HNR1.DE	2.847	4.155	6.182	6.925	6.858	7.216	7.790	8.954	5.026	6.846	5.865
P911.DE	1.924	2.271	4.083	3.852	3.746	3.963	2.917	4.212	4.182	4.170	3.228
FRE.DE	1.558	1.769	2.281	2.987	4.201	2.736	3.540	1.957	2.731	2.814	2.232
CON.DE	3.394	5.851	9.970	8.447	10.909	5.843	8.738	10.119	8.416	7.206	5.148
IFX.DE	0.783	0.957	1.609	1.511	1.608	2.220	2.535	1.678	1.195	1.632	1.134
DTE.DE	0.356	0.588	1.045	0.750	1.075	0.801	0.989	1.021	1.161	0.751	0.990
SIE.DE	3.374	3.817	5.890	6.573	6.227	7.825	6.592	4.914	5.681	6.008	4.659
DBK.DE	0.316	0.322	0.690	0.710	0.554	0.621	0.666	0.691	0.678	0.622	0.573
AIR.DE	2.974	2.849	5.358	4.604	5.261	5.466	5.359	6.175	4.295	4.941	3.872
HEI.DE	1.010	1.548	2.552	2.023	2.033	2.223	2.501	2.100	2.363	1.759	2.227
BAYN.DE	1.046	1.348	1.811	2.704	2.239	1.193	2.586	1.672	1.506	1.575	2.058
EOAN.DE	0.127	0.165	0.312	0.331	0.296	0.310	0.276	0.272	0.315	0.245	0.271
BMW.DE	2.028	2.415	4.228	5.640	2.866	6.482	8.425	4.058	3.951	5.132	2.396
RWE.DE	1.638	1.114	2.590	2.092	3.742	4.494	3.944	2.203	2.568	3.091	2.838
VOW3.DE	2.736	2.595	4.499	5.782	5.085	3.871	4.483	3.515	3.773	4.255	3.343
DTG.DE	0.611	0.866	1.532	1.861	2.012	1.573	2.019	1.420	1.382	1.339	1.484
VNA.DE	0.838	1.082	1.363	1.667	1.973	1.871	1.843	1.887	1.496	1.730	1.256
ENR.DE	1.360	1.219	1.812	2.548	3.548	1.419	2.047	1.859	1.829	1.488	1.462
SY1.DE	2.192	3.326	3.934	3.211	3.377	2.597	4.711	3.666	3.970	3.277	3.721
BEI.DE	1.075	1.273	2.523	1.697	2.502	2.909	3.343	1.864	1.870	2.412	2.727
ADS.DE	8.306	7.885	15.422	17.846	20.581	16.820	19.977	14.188	12.665	12.600	15.134
SHL.DE	0.997	1.167	1.507	1.592	1.852	3.416	2.053	2.011	1.583	2.080	1.506
ZAL.DE	1.713	2.142	5.015	4.276	4.324	6.137	4.611	4.678	3.458	3.021	3.701
DB1.DE	3.239	6.484	8.660	6.447	11.486	9.353	5.606	10.330	11.718	8.125	8.043
1COV.DE	0.875	1.048	1.727	1.828	1.651	5.206	2.330	1.940	1.724	2.117	1.406
BAS.DE	1.119	1.032	1.693	1.649	1.699	1.961	2.233	1.321	1.645	1.473	1.795
MTX.DE	3.912	4.311	7.669	7.254	9.874	9.065	7.884	8.197	8.340	10.156	6.190
MRK.DE	3.093	4.003	7.786	5.539	5.503	6.386	7.766	5.682	7.270	6.085	7.247

MAPE criteria. LSTM-GA follows it.

Table 10 introduces the comparison of the algorithms' performance using the out-of-sample R2 score, a statistical measure indicating the proportion of variance in the dependent variable that the independent variables can explain. The out-of-sample R2 score ranges from -∞ to 1, where a higher value signifies a better fit of the model to the data. Analyzing the out-of-sample R2 scores for the evaluated algorithms, it's evident that LSTM-SCSO consistently yields higher out-of-sample R2

Table 9
MAPE comparison of the algorithms.

MAPE											
Ticker	LSTM-SCO	LSTM-GA	LSTM-1D	LSTM-2D	LSTM-3D	ANN	ARIMA	GBT	DeepAR	N-BEATS	N-HITS
ALV.DE	0.024	0.027	0.029	0.040	0.037	0.069	0.063	0.067	0.035	0.037	0.047
HNR1.DE	0.018	0.026	0.039	0.044	0.044	0.046	0.050	0.057	0.033	0.044	0.037
P911.DE	0.020	0.023	0.042	0.040	0.039	0.041	0.030	0.043	0.043	0.043	0.033
FRE.DE	0.064	0.072	0.093	0.123	0.173	0.106	0.144	0.080	0.112	0.115	0.092
CON.DE	0.061	0.106	0.181	0.153	0.198	0.100	0.157	0.183	0.152	0.131	0.093
IFX.DE	0.028	0.035	0.058	0.055	0.058	0.080	0.091	0.061	0.043	0.059	0.041
DTE.DE	0.021	0.035	0.061	0.044	0.063	0.047	0.058	0.060	0.068	0.044	0.058
SIE.DE	0.030	0.034	0.053	0.059	0.056	0.068	0.059	0.044	0.050	0.054	0.042
DBK.DE	0.032	0.034	0.070	0.073	0.059	0.064	0.070	0.071	0.070	0.064	0.059
AIR.DE	0.028	0.028	0.051	0.044	0.051	0.053	0.052	0.059	0.041	0.047	0.038
HEI.DE	0.021	0.033	0.053	0.043	0.044	0.047	0.054	0.046	0.051	0.039	0.047
BAYN.DE	0.019	0.025	0.034	0.049	0.041	0.022	0.047	0.031	0.028	0.029	0.038
EOAN.DE	0.015	0.019	0.035	0.038	0.034	0.035	0.032	0.032	0.036	0.028	0.032
BMW.DE	0.030	0.035	0.062	0.082	0.042	0.096	0.122	0.059	0.057	0.074	0.035
RWE.DE	0.043	0.030	0.069	0.055	0.099	0.120	0.105	0.058	0.068	0.082	0.075
VOW3.DE	0.027	0.026	0.045	0.057	0.051	0.039	0.045	0.035	0.038	0.042	0.033
DTG.DE	0.025	0.036	0.064	0.078	0.085	0.065	0.084	0.059	0.058	0.055	0.062
VNA.DE	0.034	0.044	0.055	0.070	0.083	0.075	0.074	0.077	0.059	0.072	0.051
ENR.DE	0.095	0.086	0.128	0.180	0.249	0.098	0.143	0.131	0.128	0.105	0.102
SY1.DE	0.021	0.032	0.038	0.031	0.032	0.025	0.045	0.035	0.038	0.031	0.036
BEI.DE	0.011	0.013	0.026	0.018	0.026	0.030	0.034	0.020	0.020	0.025	0.028
ADS.DE	0.061	0.057	0.111	0.132	0.154	0.117	0.146	0.102	0.093	0.092	0.110
SHL.DE	0.021	0.024	0.031	0.033	0.038	0.071	0.043	0.042	0.033	0.042	0.032
ZAL.DE	0.059	0.075	0.182	0.153	0.157	0.199	0.159	0.166	0.120	0.100	0.126
DB1.DE	0.021	0.041	0.055	0.041	0.073	0.060	0.036	0.066	0.075	0.052	0.051
1COV.DE	0.024	0.028	0.046	0.051	0.046	0.147	0.062	0.052	0.044	0.059	0.039
BAS.DE	0.026	0.025	0.041	0.040	0.041	0.047	0.053	0.032	0.039	0.035	0.042
MTX.DE	0.021	0.023	0.041	0.039	0.053	0.049	0.043	0.044	0.045	0.055	0.034
MRK.DE	0.018	0.023	0.045	0.032	0.032	0.037	0.045	0.033	0.042	0.035	0.042

Table 10
Out of sample R2 comparison of the algorithms.

Out-of-sample R2											
Ticker	LSTM-SCO	LSTM-GA	LSTM-1D	LSTM-2D	LSTM-3D	ANN	ARIMA	GBT	DeepAR	N-BEATS	N-HITS
ALV.DE	0.858	0.829	0.794	0.617	0.694	0.089	0.086	0.139	0.708	0.690	0.480
HNR1.DE	0.948	0.887	0.738	0.691	0.711	0.643	0.633	0.508	0.847	0.698	0.753
P911.DE	0.699	0.575	−0.370	−0.209	−0.146	−0.286	0.271	−0.452	−0.424	−0.424	0.147
FRE.DE	0.810	0.766	0.592	0.310	−0.331	0.513	0.071	0.707	0.439	0.400	0.609
CON.DE	0.845	0.568	−0.270	0.080	−0.501	0.493	0.058	−0.272	0.095	0.307	0.627
IFX.DE	0.926	0.884	0.688	0.718	0.699	0.497	0.252	0.669	0.830	0.684	0.840
DTE.DE	0.899	0.726	0.167	0.534	0.112	0.414	0.215	0.183	−0.031	0.548	0.244
SIE.DE	0.888	0.862	0.654	0.612	0.657	0.422	0.598	0.767	0.705	0.666	0.786
DBK.DE	0.921	0.912	0.606	0.603	0.746	0.649	0.635	0.615	0.615	0.690	0.723
AIR.DE	0.760	0.763	0.298	0.404	0.277	0.148	0.160	−0.047	0.497	0.318	0.564
HEI.DE	0.937	0.862	0.641	0.751	0.758	0.657	0.612	0.709	0.662	0.790	0.723
BAYN.DE	0.932	0.881	0.793	0.556	0.671	0.903	0.561	0.824	0.859	0.847	0.727
EOAN.DE	0.974	0.961	0.860	0.839	0.870	0.856	0.887	0.889	0.859	0.907	0.891
BMW.DE	0.768	0.693	0.036	−0.529	0.488	−1.620	−2.464	0.178	0.142	−0.328	0.668
RWE.DE	0.326	0.676	−0.615	−0.151	−2.253	−3.598	−2.675	−0.295	−0.760	−1.240	−0.905
VOW3.DE	0.885	0.881	0.636	0.506	0.613	0.730	0.633	0.778	0.739	0.716	0.820
DTG.DE	0.918	0.825	0.469	0.250	0.162	0.414	0.074	0.550	0.567	0.607	0.485
VNA.DE	0.968	0.943	0.909	0.861	0.805	0.836	0.840	0.832	0.895	0.856	0.922
ENR.DE	0.713	0.760	0.452	−0.026	−0.866	0.662	0.287	0.440	0.469	0.627	0.631
SY1.DE	0.671	0.309	−0.095	0.281	0.211	0.480	−0.547	0.064	−0.146	0.247	0.048
BEI.DE	0.952	0.936	0.740	0.876	0.759	0.656	0.557	0.855	0.856	0.767	0.703
ADS.DE	0.932	0.938	0.774	0.674	0.559	0.728	0.617	0.804	0.845	0.847	0.778
SHL.DE	0.918	0.885	0.791	0.781	0.718	−0.052	0.619	0.646	0.780	0.668	0.775
ZAL.DE	0.978	0.965	0.825	0.859	0.858	0.667	0.841	0.838	0.907	0.927	0.893
DB1.DE	0.750	0.146	−0.539	0.075	−1.645	−0.868	0.187	−1.162	−1.722	−0.455	−0.391
1COV.DE	0.967	0.955	0.872	0.847	0.884	−0.180	0.775	0.844	0.878	0.798	0.912
BAS.DE	0.935	0.942	0.846	0.850	0.845	0.795	0.731	0.906	0.858	0.880	0.824
MTX.DE	0.890	0.865	0.575	0.615	0.316	0.347	0.546	0.520	0.497	0.251	0.720
MRK.DE	0.808	0.685	−0.217	0.397	0.408	0.119	−0.141	0.323	−0.080	0.259	−0.051

scores across multiple stock tickers, indicating their superior ability to capture and explain the variance in stock price movements. These findings reiterate the prominence of LSTM-SCSO in capturing and explaining stock price fluctuations, reinforcing their suitability for robust and accurate stock market predictions. However, it's essential to consider the context of each stock's unique behavior, which can impact the performance of different algorithms. LSTM-SCSO model performs the best result. It is the best algorithm for the stock market price prediction dataset for out-of-sample R2 criteria. LSTM-GA follows it.

The provided tables, encompassing the MSE, MAE, MAPE, and out-of-sample R2 score comparisons of various algorithms across different stock tickers, offer a comprehensive perspective on the algorithms' performances in predicting stock price movements. These evaluations shed light on the algorithms' strengths and limitations, contributing to a broader understanding of their applicability in the complex and dynamic domain of financial forecasting.

Upon examining the MSE, MAE, MAPE, and out-of-sample R2 score tables, it becomes evident that LSTM-SCSO outperforms other algorithms, yielding lower error values across most of the considered stock tickers. This suggests that these two algorithms are capable of making more accurate predictions, as they exhibit better alignment between predicted and actual stock prices.

In summary, analysis of the provided tables underscores the prominence of LSTM-SCSO as the most successful algorithm for predicting stock prices. Their ability to consistently achieve lower error metrics, higher accuracy, and better explanatory power highlights their potential for making more reliable and insightful stock market forecasts. However, it's important to acknowledge that algorithm performance can be influenced by various factors, including the specific characteristics of individual stocks, market conditions, and the quality of input data. Therefore, while LSTM-SCSO emerges as the top contender, a thorough understanding of its strengths and weaknesses in different contexts is crucial for making informed decisions in real-world financial applications.

The analysis of the DAX stock dataset from 2018 to 2023 using various algorithms (ANN, LSTM1D, LSTM2D, LSTM3D, LSTM-GA, Auto-ARIMA, GBT, DeepAR, N-BEATS, N-HITS and LSTM-SCSO) reveals consistent patterns across all evaluation metrics (MSE, MAE, MAPE, and out of sample R2 score).

The LSTM-SCSO algorithm consistently outperforms other methods across most stock tickers, demonstrating superior predictive accuracy and explanatory power. This performance is evident in:

1. Lower error rates: LSTM-SCSO consistently achieves the lowest MSE, MAE, and MAPE values, indicating more accurate predictions with smaller deviations from actual stock prices.
2. Higher explanatory power: The consistently higher out-of-sample R2 scores for LSTM-SCSO suggest that this model captures a larger proportion of the variance in stock price movements.

The superior performance of LSTM-SCSO can be attributed to.

1. Effective temporal dependency capture: The LSTM architecture excels at learning long-term dependencies in time series data.
2. Optimized hyperparameters: The SCSO algorithm likely finds a more optimal set of hyperparameters, enhancing the LSTM's performance.
3. Adaptive learning: The combination of LSTM and SCSO may allow the model to adapt better to the complex, non-linear nature of stock price movements.

While LSTM-SCSO shows the best overall performance, it's important to note that the second-best performer is typically LSTM-GA, highlighting the effectiveness of evolutionary optimization techniques in this context.

These results underscore the potential of hybrid models that combine deep learning architectures with advanced optimization algorithms for stock price prediction. However, it's crucial to consider that performance may vary depending on specific stock characteristics and market conditions.

4.4. *Model confidence set*

4.4.1. *MSE*

The Model Confidence Set (MCS) procedure was implemented to statistically evaluate and rank the performance of all models. The test was conducted with 5000 bootstrap replications and a significance level of $\alpha = 0.05$. The elimination process began with all 11 models and proceeded iteratively based on mean MSE values. The models were eliminated in the following order: LSTM-3D (mean MSE: 56.83), ARIMA (56.08), ANN (53.35), GBT (44.99), LSTM-2D (44.14), LSTM-1D (41.06), DeepAR (36.59), N-BEATS (36.17), N-HITS (34.10), and LSTM-GA (14.55). The LSTM-SCSO model emerged as the sole superior model in the final set, with a mean MSE of 10.58 across all stocks. The sequential elimination pattern reveals a clear hierarchy in model performance, with traditional models being eliminated in earlier iterations, while more sophisticated hybrid approaches demonstrated greater predictive accuracy. The LSTM-SCSO's survival as the only model in the final confidence set provides strong statistical evidence for its superior predictive capabilities compared to all other tested models. Table 11 shows the model confidence set elimination process for MSE.

Table 12 shows the final model confidence set statistics for MSE.

4.4.2. *MAE*

The Model Confidence Set (MCS) procedure was applied to evaluate the MAE performance of all models with 5000 bootstrap replications and $\alpha = 0.05$ significance level. The elimination process started with 11 models and proceeded iteratively based on mean MAE values. The models were eliminated in the following sequence: ARIMA (mean MAE: 4.79), ANN (4.69), LSTM-3D (4.61), GBT (4.29), LSTM-2D (4.12), LSTM-1D (4.10), N-BEATS (3.91), DeepAR (3.90), N-HITS (3.61), and LSTM-GA (2.50). The LSTM-SCSO model emerged as the sole superior model with a mean MAE of 2.06 across all stocks. The sequential elimination pattern demonstrates a clear hierarchy in predictive accuracy, with traditional models being eliminated earlier, while the LSTM-SCSO maintained consistently lower error rates. The survival of LSTM-SCSO as the only model in the final confidence set provides robust statistical evidence of its superior predictive capabilities in terms of absolute error measures. Table 13 shows the model confidence set elimination process for MAE.

Table 14 shows the final model confidence set statistics for MAE.

4.4.3. *MAPE*

The Model Confidence Set procedure was implemented to evaluate model performance using MAPE values across 11 competing models. The analysis was conducted with 5000 bootstrap replications at $\alpha = 0.05$ significance level. The elimination process revealed a clear hierarchy in model performance. The LSTM-3D model was eliminated first with a mean MAPE of 0.0744, followed by ARIMA (0.0740), ANN (0.0708), LSTM-2D (0.0654), GBT (0.0636), LSTM-1D (0.0634), DeepAR (0.0593), N-BEATS (0.0583), N-HITS (0.0536), and LSTM-GA (0.0378). The LSTM-SCSO model emerged as the sole superior model with a mean MAPE of 0.0317, demonstrating significantly better predictive accuracy. The superior model's performance is characterized by low variability (standard deviation: 0.0189) and consistent accuracy across different stocks, with MAPE values ranging from 0.011 to 0.095. Table 15 shows

Table 11
Model confidence set elimination process (MSE).

Iteration	Eliminated Model	Mean MSE	Models Remaining	Elimination Order
1	LSTM-3D	56.834	10	1st
2	ARIMA	56.083	9	2nd
3	ANN	53.350	8	3rd
4	GBT	44.998	7	4th
5	LSTM-2D	44.141	6	5th
6	LSTM-1D	41.060	5	6th
7	DeepAR	36.591	4	7th
8	N-BEATS	36.166	3	8th
9	N-HITS	34.102	2	9th
10	LSTM-GA	14.549	1	10th

Table 12
Final model confidence set statistics (MSE).

Statistic	LSTM-SCO (Superior Model)
Mean MSE	10.582
Standard Deviation	19.767
Minimum MSE	0.025
25th Percentile	1.487
Median MSE	3.584
75th Percentile	14.237
Maximum MSE	105.623
Number of Stocks	29

Table 13
Model confidence set elimination process (MAE).

Iteration	Eliminated Model	Mean MAE	Models Remaining	Elimination Order
1	ARIMA	4.793	10	1st
2	ANN	4.688	9	2nd
3	LSTM-3D	4.606	8	3rd
4	GBT	4.289	7	4th
5	LSTM-2D	4.120	6	5th
6	LSTM-1D	4.100	5	6th
7	N-BEATS	3.914	4	7th
8	DeepAR	3.898	3	8th
9	N-HITS	3.615	2	9th
10	LSTM-GA	2.496	1	10th

Table 14
Final model confidence set statistics (MAE).

Statistic	LSTM-SCO (Superior Model)
Mean MAE	2.063
Standard Deviation	1.671
Minimum MAE	0.127
25th Percentile	0.997
Median MAE	1.638
75th Percentile	2.974
Maximum MAE	8.306
Number of Stocks	29

Table 15
Model confidence set elimination process (MAPE).

Iteration	Eliminated Model	Mean MAPE	Models Remaining	Elimination Order
1	LSTM-3D	0.0744	10	1st
2	ARIMA	0.0740	9	2nd
3	ANN	0.0708	8	3rd
4	LSTM-2D	0.0654	7	4th
5	GBT	0.0636	6	5th
6	LSTM-1D	0.0634	5	6th
7	DeepAR	0.0593	4	7th
8	N-BEATS	0.0583	3	8th
9	N-HITS	0.0536	2	9th
10	LSTM-GA	0.0378	1	10th

Table 16
Final model confidence set statistics (MAPE).

Statistic	LSTM-SCO (Superior Model)
Mean MAPE	0.0317
Standard Deviation	0.0189
Minimum MAPE	0.0110
25th Percentile	0.0210
Median MAPE	0.0250
75th Percentile	0.0320
Maximum MAPE	0.0950
Number of Stocks	29

the model confidence set elimination process for MAPE.

Table 16 shows the final model confidence set statistics for MAPE.

4.4.4. *Out of sample R2*

The Model Confidence Set procedure was implemented to evaluate model performance using out-of-sample R2 values across 11 competing models. The analysis was conducted with 5000 bootstrap replications at $\alpha = 0.05$ significance level. The elimination process revealed a clear hierarchy in model performance. ARIMA was eliminated first with a mean R2 of 0.2041, followed by ANN (0.2108), LSTM-3D (0.2786), GBT (0.4262), DeepAR (0.4271), LSTM-1D (0.4362), N-BEATS (0.4672), LSTM-2D (0.4714), N-HITS (0.5499), and LSTM-GA (0.7890). The LSTM-SCSO model emerged as the sole superior model with a mean R2 of 0.8543, demonstrating significantly better explanatory power. The superior model's performance is characterized by strong consistency (standard deviation: 0.1350) and high predictive power across different stocks, with R2 values ranging from 0.326 to 0.978. Table 17 shows the model confidence set elimination process for out-of-sample R2.

Table 18 shows the final model confidence set statistics for out-of-sample R2.

5. Economic implications and practical applications

The economic value of the LSTM-SCSO model was evaluated through a comprehensive portfolio analysis comparing its performance against the DAX index. A trading strategy was implemented using the LSTM-SCSO predictions for the test period, with portfolio rebalancing based on the model's forecasts. The results demonstrate significant economic implications for practical applications.

The LSTM-SCSO portfolio achieved an annualized return of 66.25%, substantially outperforming the DAX index's return of 47.45%. This superior return performance indicates the model's ability to identify profitable trading opportunities. The enhanced return was accompanied by higher annualized volatility of 15.68% compared to the DAX index's 10.94%, reflecting the more active trading approach based on model predictions.

Risk-adjusted performance metrics provide additional insights into the economic value of the LSTM-SCSO model. The portfolio achieved a Sharpe ratio of 2.9091, slightly lower than the DAX index's 3.0714, suggesting comparable risk-adjusted returns. The maximum drawdown for the LSTM-SCSO portfolio was −7.79%, moderately higher than the DAX index's −5.00%, indicating slightly increased downside risk.

These results hold several practical implications for investment applications. The model demonstrates its capability to generate excess returns through active trading signals, though with increased volatility. The comparable Sharpe ratios suggest that the additional returns adequately compensate for the increased risk. The moderate maximum drawdown indicates acceptable risk control despite the active trading approach.

These findings validate the economic value of the LSTM-SCSO model beyond mere predictive accuracy, demonstrating its potential utility in

Table 17
Model Confidence Set Elimination Process (Out of sample R2).

Iteration	Eliminated Model	Mean R2	Models Remaining	Elimination Order
1	ARIMA	0.2041	10	1st
2	ANN	0.2108	9	2nd
3	LSTM-3D	0.2786	8	3rd
4	GBT	0.4262	7	4th
5	DeepAR	0.4271	6	5th
6	LSTM-1D	0.4362	5	6th
7	N-BEATS	0.4672	4	7th
8	LSTM-2D	0.4714	3	8th
9	N-HITS	0.5499	2	9th
10	LSTM-GA	0.7890	1	10th

Table 18
Final Model Confidence Set Statistics (Out of sample R2).

Statistic	LSTM-SCO (Superior Model)
Mean R2	0.8543
Standard Deviation	0.1350
Minimum R2	0.3260
25th Percentile	0.8080
Median R2	0.8990
75th Percentile	0.9350
Maximum R2	0.9780
Number of Stocks	29

real-world trading applications. The model's ability to generate substantial excess returns while maintaining reasonable risk metrics suggests its viability as a practical tool for investment management.

The results should be interpreted within the context of the test period conditions. Further research could explore the strategy's performance across different market regimes and its scalability to larger portfolios. Additionally, the integration of position sizing and risk management overlays could potentially enhance the risk-adjusted performance metrics. Table 19 shows the comparison of the LSTM-SCSO and DAX index portfolios.

6. Conclusion

In this study, the realm of stock price prediction has been delved into by leveraging various artificial intelligence algorithms, with a particular focus on the novel LSTM-SCSO approach. The objective was to enhance the accuracy of predictions and provide valuable insights for traders and investors seeking to navigate the complexities of the stock market.

A comprehensive review of the literature reveals a rich landscape of approaches to stock price prediction, ranging from traditional time series methods to advanced deep learning techniques. It has been observed that the field has seen significant advancements in recent years, with increasingly sophisticated models being explored that combine multiple techniques and data sources to improve prediction accuracy.

The application of SCSO to optimize the LSTM model introduced a novel and promising approach to refining the predictive capabilities of the algorithm. This aligns with the broader trend in the field towards hybrid and ensemble models, which have shown superior performance compared to single-algorithm approaches. The analysis of the DAX stock dataset from 2018 to 2023 using various algorithms (LSTM1D, LSTM2D, LSTM3D, ANN, LSTM-GA, Auto-ARIMA, GBT, DeepAR, N-BEATS, N-HITS, and LSTM-SCSO) revealed noteworthy patterns in algorithm performance across multiple metrics, such as MSE, MAE, MAPE, and out of sample R2 scores.

Among the considered algorithms, LSTM-SCSO emerged as the most successful in consistently achieving superior results. Its ability to outperform other algorithms in terms of accuracy, error reduction, and explanatory power underscores its potential as a robust tool for stock price prediction. This aligns with findings from other researchers who have reported success with optimized LSTM models and hybrid approaches.

The Model Confidence Set analysis, conducted with 5000 bootstrap replications at $\alpha = 0.05$ significance level, provided robust statistical validation of LSTM-SCSO's superior performance. Through sequential elimination across all evaluation metrics (MSE, MAE, MAPE, out-of-sample R2), LSTM-SCSO emerged as the sole superior model. For MSE, the model achieved a mean of 10.582 compared to the next best performer, LSTM-GA, at 14.549. Similar patterns were observed in MAE (2.063 vs 2.496) and MAPE (0.0317 vs 0.0378). The out-of-sample R2 analysis demonstrated LSTM-SCSO's exceptional explanatory power with a mean of 0.8543, significantly outperforming other models. This comprehensive statistical evaluation establishes LSTM-SCSO's predictive superiority with high confidence.

The economic value of LSTM-SCSO was validated through portfolio analysis against the DAX index. The LSTM–SCSO–based portfolio achieved an annualized return of 66.25%, substantially outperforming the DAX index's 47.45%. While this enhanced return was accompanied by higher volatility (15.68% vs 10.94%), the risk-adjusted performance remained strong with a Sharpe ratio of 2.9091, comparable to the DAX index's 3.0714. The maximum drawdown was maintained at −7.79%, indicating effective risk management despite the active trading approach. These results demonstrate that LSTM-SCSO's superior predictive accuracy translates into meaningful economic value, though users should remain cognizant of the increased volatility inherent in model-driven trading strategies.

Table 19
Comparison of the LSTM-SCSO and DAX index.

Portfolio Name	Annualized Return	Annualized Volatility	Sharpe Ratio	Maximum Drawdown
LSTM-SCSO	0.6625	0.1568	2.9091	−0.0779
DAX index	0.4745	0.1094	3.0714	−0.05

The future of stock price prediction appears promising with the continued advancement of hybrid AI models like LSTM-SCSO. The demonstrated success of this approach, both in statistical accuracy and economic value generation, opens new avenues for further research and practical applications. The integration of additional data sources, enhanced optimization techniques, and more sophisticated risk management frameworks could further improve the model's capabilities. As computational power increases and machine learning techniques evolve, the potential for more precise and reliable stock market predictions grows. This research establishes a strong foundation for future developments in financial forecasting, offering hope for more stable and profitable investment strategies. The LSTM-SCSO model represents a significant step forward in bridging the gap between academic research and practical trading applications, suggesting a future where AI-driven investment decisions become increasingly reliable and profitable while maintaining appropriate risk controls.

Data availability and access

The data is collected from Yahoo Finance (https://finance.yahoo.com). The data is open and accessible at Yahoo Finance.

Declaration of generative AI in scientific writing

Gpt-4o and Grammarly were used for grammar checking in the article.

References

Almaafi, A., Bajaba, S., & Alnori, F. (2023). Stock price prediction using ARIMA versus XGBoost models: The case of the largest telecommunication company in the Middle East. *International Journal of Information Technology (Singapore), 15*(4), 1813–1818. https://doi.org/10.1007/s41870-023-01260-4

Almasarweh, M., & Alwadi, S. (2018). ARIMA model in predicting banking stock market data. *Modern Applied Science, 12*(11), 309.

Banke, M., Lenger, S., & Pott, C. (2022). ESG ratings in the corporate reporting of DAX40 companies in Germany: Effects on market participants. *Sustainability, 14*(15), 9742. https://doi.org/10.3390/su14159742

Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (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

Bazrkar, M. J., & Hosseini, S. (2023). Predict stock prices using supervised learning algorithms and particle swarm optimization algorithm. *Computational Economics, 62* (1), 165–186. https://doi.org/10.1007/s10614-022-10273-3. Scopus.

Bi, J.-W., Li, H., & Fan, Z.-P. (2021). Tourism demand forecasting with time series imaging: A deep learning model. *Annals of Tourism Research, 90*, Article 103255. https://doi.org/10.1016/j.annals.2021.103255

Bose, A., Hsu, C.-H., Roy, S. S., Lee, K. C., Mohammadi-ivatloo, B., & Abimannan, S. (2021). Forecasting stock price by hybrid model of cascading multivariate adaptive regression Splines and deep neural network. *Computers & Electrical Engineering, 95*. https://doi.org/10.1016/j.compeleceng.2021.107405

Bühler, W., & Kempf, A. (1995). Dax index futures: Mispricing and arbitrage in German markets. *Journal of Futures Markets, 15*(7), 833–859. https://doi.org/10.1002/fut.3990150706

Challu, C., Olivares, K. G., Oreshkin, B. N., Ramirez, F. G., Canseco, M. M., & Dubrawski, A. (2023). Nhits: Neural hierarchical interpolation for time series forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence, 37*(6). https://doi.org/10.1609/aaai.v37i6.25854. Article 6.

Chandar, S. K. (2021). Forecasting intraday stock price using ANFIS and bio-inspired algorithms. *International Journal of Networking and Virtual Organisations, 25*(1), 29–47. https://doi.org/10.1504/IJNVO.2021.117754. Scopus.

Chen, X., Cao, L., Cao, Z., & Zhang, H. (2024). A multi-feature stock price prediction model based on multi-feature calculation, LASSO feature selection, and Ca-LSTM network. *Connection Science, 36*(1). https://doi.org/10.1080/09540091.2023.2286188

Chiew, E., & Choong, S. S. (2022). A solution for M5 Forecasting - uncertainty: Hybrid gradient boosting and autoregressive recurrent neural network for quantile estimation. *International Journal of Forecasting, 38*(4), 1442–1447. https://doi.org/10.1016/j.ijforecast.2022.01.009

Chopra, R., & Sharma, G. D. (2021). Application of artificial intelligence in stock market forecasting: A critique, review, and research agenda. *Journal of Risk and Financial Management, 14*(11), 526. https://doi.org/10.3390/jrfm14110526

Dengiz, B., Alabas-Uslu, C., & Dengiz, O. (2009). A tabu search algorithm for the training of neural networks. *Journal of the Operational Research Society, 60*(2), 282–291.

Duan, H., Huang, J., Wang, S., & He, C. (2021). A novel grey model of impulse delay and its application in forecasting stock price. *Journal of Intelligent and Fuzzy Systems, 41*(2), 3395–3413. https://doi.org/10.3233/JIFS-210726. Scopus.

Dwivedi, V. K., & Gore, M. M. (2021). A historical data based ensemble system for efficient stock price prediction. *Recent Advances in Computer Science and Communications, 14*(4), 1182–1212. https://doi.org/10.2174/2213275912666190730161807

Feng, L., Qi, J., & Lucey, B. (2024). Enhancing cryptocurrency market volatility forecasting with daily dynamic tuning strategy. *International Review of Financial Analysis, 94*, Article 103239. https://doi.org/10.1016/j.irfa.2024.103239

Fu, T., Huang, D., Feng, L., & Tang, X. (2024). More is better? The impact of predictor choice on the INE oil futures volatility forecasting. *Energy Economics, 134*, Article 107540. https://doi.org/10.1016/j.eneco.2024.107540

Fujimoto, Y., Nakagawa, K., Imajo, K., & Minami, K. (2022). *Uncertainty aware trader-company method: Interpretable stock price prediction capturing uncertainty*. 1238–1245. Scopus. https://doi.org/10.1109/BigData55660.2022.10021096.

Gao, Y., Wang, R., & Zhou, E. (2021). *Stock prediction based on optimized LSTM and GRU models*. Scientific Programming. https://doi.org/10.1155/2021/4055281, 2021.

Gülcü, Ş. (2022). Training of the feed forward artificial neural networks using dragonfly algorithm. *Applied Soft Computing, 124*, Article 109023. https://doi.org/10.1016/j.asoc.2022.109023

Gülmez, B. (2023a). A novel deep learning model with the Grey Wolf Optimization algorithm for cotton disease detection. *Journal of Universal Computer Science, 29*(29), 595–626. https://doi.org/10.3897/jucs.94183.

Gülmez, B. (2023b). A novel deep neural network model based Xception and genetic algorithm for detection of COVID-19 from X-ray images. *Annals of Operations Research, 328*(1), 617–641. https://doi.org/10.1007/s10479-022-05151-y.

Gülmez, B. (2023c). Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. *Expert Systems with Applications, 227*, 120346. https://doi.org/10.1016/j.eswa.2023.120346.

Gülmez, B. (2024a). A comprehensive review of convolutional neural networks based disease detection strategies in potato agriculture. *Potato Research*. https://doi.org/10.1007/s11540-024-09786-1.

Gülmez, B. (2024b). A new multi-objective hyperparameter optimization algorithm for COVID-19 detection from x-ray images. *Soft Computing, 28*(19), 11601–11617. https://doi.org/10.1007/s00500-024-09872-z.

Gülmez, B. (2024c). Advancements in maize disease detection: A comprehensive review of convolutional neural networks. *Computers in Biology and Medicine, 183*, 109222. https://doi.org/10.1016/j.compbiomed.2024.109222.

Gülmez, B. (2024d). Advancements in rice disease detection through convolutional neural networks: A comprehensive review. *Heliyon, 10*(12), Article e33328. https://doi.org/10.1016/j.heliyon.2024.e33328.

Gülmez, B., Emmerich, M., & Fan, Y. (2024). Multi-objective optimization for green delivery routing problems with flexible time windows. *Applied Artificial Intelligence, 38*(1), 2325302. https://doi.org/10.1080/08839514.2024.2325302.

Gülmez, B., & Kulluk, S. (2019). Social spider algorithm for training artificial neural networks. *International Journal of Business Analytics (IJBAN), 6*(4), 32–49. https://doi.org/10.4018/IJBAN.2019100103.

Gülmez, B., & Kulluk, S. (2023a). Analysis and price prediction of secondhand vehicles in Türkiye with big data and machine learning techniques. *Journal of the Faculty of Engineering and Architecture of Gazi University, 38*(4), 2279–2290. https://doi.org/10.17341/gazimmfd.980840.

Hamilton, J. D. (2020). *Time series analysis*. Princeton University Press.

Hellström, T., & Holmström, K. (1998). *Predicting the stock market*.

Henne, A., Ostrowski, S., & Reichling, P. (2009). Dividend yield and stability versus performance on the German stock market: A descriptive study. *Review of Managerial Science, 3*(3), 225–248. https://doi.org/10.1007/s11846-009-0029-2

Ilyas, Q. M., Iqbal, K., Ijaz, S., Mehmood, A., & Bhatia, S. (2022). A hybrid model to predict stock closing price using novel features and a fully modified hodrick–prescott filter. *Electronics (Switzerland), 11*(21). https://doi.org/10.3390/electronics11213588

Jafari, A., & Haratizadeh, S. (2022). Gcnet: Graph-based prediction of stock price movement using graph convolutional network. *Engineering Applications of Artificial Intelligence, 116*. https://doi.org/10.1016/j.engappai.2022.105452

Joel, L., Parthasarathy, S., Venkatesan, P., & Nandhini, S. (2023). IPH2O: Island parallel-harris Hawks optimizer-based CLSTM for stock price movement prediction. *Annals of Data Science*. https://doi.org/10.1007/s40745-023-00489-x

Kim, Y., & Kim, S. (2021). Forecasting charging demand of electric vehicles using time-series models. *Energies, 14*(5). https://doi.org/10.3390/en14051487. Article 5.

Kumar Chandar, S., & Punjabi, H. (2022). Cat swarm optimization algorithm tuned multilayer perceptron for stock price prediction. *Computer Assisted Mechanics and Engineering Sciences, 29*(1–2), 145–160. https://doi.org/10.24423/cames.394. Scopus.

Kumar, N., & Dadhich, M. (2014). Risk management for investors in stock market. *EXCEL International Journal of Multidisciplinary Management Studies, 4*(3), 103–108.

Lainder, A. D., & Wolfinger, R. D. (2022). Forecasting with gradient boosted trees: Augmentation, tuning, and cross-validation strategies: Winning solution to the M5 Uncertainty competition. *International Journal of Forecasting, 38*(4), 1426–1433. https://doi.org/10.1016/j.ijforecast.2021.12.003

Li, F., & Abo Keir, M. Y. (2022). Mathematical model of back propagation for stock price forecasting. *Applied Mathematics and Nonlinear Sciences, 7*(1), 523–532. https://doi.org/10.2478/amns.2021.1.00089

Li, J., Chen, W., Zhou, Z., Yang, J., & Zeng, D. (2024). DeepAR-Attention probabilistic prediction for stock price series. *Neural Computing & Applications, 36*(25), 15389–15406. https://doi.org/10.1007/s00521-024-09916-3

Li, H., Hu, Y., & Liu, H. (2023). A novel LASSO-ATT-LSTM model of stock price prediction based on multi-source heterogeneous data. *Journal of Intelligent and Fuzzy Systems, 44*(6), 10511–10521. https://doi.org/10.3233/JIFS-221919

Li, H., Hua, J., Li, J., & Li, G. (2020). Stock forecasting model FS-LSTM based on the 5G Internet of Things. *Wireless Communications and Mobile Computing, 2020*. https://doi.org/10.1155/2020/7681209

Li, G., Zhang, A., Zhang, Q., Wu, D., & Zhan, C. (2022). Pearson correlation coefficient-based performance enhancement of broad learning system for stock price prediction. *IEEE Transactions on Circuits and Systems II: Express Briefs, 69*(5), 2413–2417. https://doi.org/10.1109/TCSII.2022.3160266

Liao, H., Huang, J., & Tang, Y. (2024). Leet: Stock market forecast with longterm emotional change enhanced temporal model. *PeerJ Computer Science, 10*. https://doi.org/10.7717/peerj-cs.1969

Luo, Y.-X., & Ji, Y. (2022). Prediction of the stock adjusted closing price based on improved PSO-LSTM neural network. *Scopus, 43–48*. https://doi.org/10.1109/ICMLC56445.2022.9941330, 2022-September.

M, I., Ahmad, S., Jha, S., Alam, A., Yaseen, M., & Abdeljaber, H. A. M. (2022). A novel AI-based stock market prediction using machine learning algorithm. *Scientific Programming*, 1–11. https://doi.org/10.1155/2022/4808088, 2022.

Maguluri, L. P., & Ragupathy, R. (2020). Comparative analysis of companies stock price prediction using time series algorithm. *International Journal of Engineering Trends and Technology, 68*(11), 9–15. https://doi.org/10.14445/22315381/IJETT-V68I11P202

Mehtab, S., & Sen, J. (2020). Stock price prediction using CNN and LSTM-based deep learning models. *International Conference on Decision Aid Sciences and Application (DASA)*, 447–453, 2020.

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

Mittal, A., Kumar, P., Roy, P. P., Balasubramanian, R., & Chaudhuri, B. B. (2019). A modified LSTM model for continuous sign language recognition using leap motion. *IEEE Sensors Journal, 19*(16), 7056–7063.

Nakagawa, K., Imamura, M., & Yoshida, K. (2019). Stock price prediction using k-medoids clustering with indexing dynamic time warping. *Electronics and Communications in Japan, 102*(2), 3–8. https://doi.org/10.1002/ecj.12140

Nasios, I., & Vogklis, K. (2022). Blending gradient boosted trees and neural networks for point and probabilistic forecasting of hierarchical time series. *International Journal of Forecasting, 38*(4), 1448–1459. https://doi.org/10.1016/j.ijforecast.2022.01.001

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). *N-BEATS: Neural basis expansion analysis for interpretable time series forecasting* (arXiv:1905.10437). *arXiv*. https://doi.org/10.48550/arXiv.1905.10437

Rajanikanth, J., Haritha, K., & Shankar, R. S. (2023). Forecasting stock close price using machine learning models. *ARPN Journal of Engineering and Applied Sciences, 18*(4), 412–420.

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting, 36*(3), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

Seyyedabbasi, A., & Kiani, F. (2023). Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. *Engineering with Computers, 39*(4), 2627–2651. https://doi.org/10.1007/s00366-022-01604-x

Staffini, A. (2022). Stock price forecasting by a deep convolutional generative adversarial network. *Frontiers in Artificial Intelligence, 5*. https://doi.org/10.3389/frai.2022.837596

Stapf, J., & Werner, T. (2003). How wacky is the DAX? The changing structure of German stock market volatility. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2785248

Talavera-Llames, R., Pérez-Chacón, R., Troncoso, A., & Martínez-Álvarez, F. (2018). Big data time series forecasting based on nearest neighbours distributed computing with Spark. *Knowledge-Based Systems, 161*, 12–25. https://doi.org/10.1016/j.knosys.2018.07.026

Tang, Y. M., Chau, K.-Y., Li, W., & Wan, T. W. (2020). Forecasting economic recession through share price in the logistics industry with artificial intelligence (AI). *Computation, 8*(3). https://doi.org/10.3390/COMPUTATION8030070

Vanstone, B. J., Gepp, A., & Harris, G. (2019). Do news and sentiment play a role in stock price prediction? *Applied Intelligence, 49*(11), 3815–3820. https://doi.org/10.1007/s10489-019-01458-9

Wang, P., Li, X., Qin, Z., Qu, Y., & Zhang, Z. (2022). Stock price forecasting based on wavelet filtering and ensembled machine learning model. *Mathematical Problems in Engineering, 2022*. https://doi.org/10.1155/2022/4024953

Xiao, D., & Su, J. (2022). *Research on stock price time series prediction based on deep learning and autoregressive integrated moving average*. Scientific Programming. https://doi.org/10.1155/2022/4758698, 2022.

Yu, Z., Qin, L., Chen, Y., & Parmar, M. D. (2020). Stock price forecasting based on LLE-BP neural network model. *Physica A: Statistical Mechanics and its Applications, 553*. https://doi.org/10.1016/j.physa.2020.124197

Yuan, X., He, C., Xu, H., & Sun, Y. (2023). Particle swarm optimization LSTM based stock prediction model. *Scopus*, 513–516. https://doi.org/10.1109/ACCTCS58815.2023.00104

Zhang, Q., Lu, M., & Dai, L. (2022). On mixed model for improvement in stock price forecasting. *Computer Systems Science and Engineering, 41*(2), 795–809. https://doi.org/10.32604/csse.2022.019987

Zhang, Q., Zhang, Y., Bao, F., Zhang, C., & Liu, P. (2024). Smpdf: Stock movement prediction based on stock prices and text. *International Journal of General Systems, 53*(4), 509–526. https://doi.org/10.1080/03081079.2023.2294132

Zhu, W., Dai, W., Tang, C., Zhou, G., Liu, Z., & Zhao, Y. (2024). PMANet: A time series forecasting model for Chinese stock price prediction. *Scientific Reports, 14*(1). https://doi.org/10.1038/s41598-024-69303-9