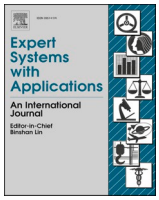
Expert Systems With Applications 227 (2023) 120346

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm

Burak Gülmez [a,b,*]

[a] Leiden Institute of Advanced Computer Science, Leiden University, Leiden, the Netherlands
[b] Mine Apt, Altay Mah. Sehit A. Taner Ekici Sk. No: 6, 06820, Etimesgut, Ankara, Türkiye

ARTICLE INFO

Keywords:
Artificial intelligence
Artificial rabbits optimization algorithm
Deep learning
LSTM
Stock price prediction

ABSTRACT

The stock market is a financial market where shares of publicly listed corporations are purchased and sold. It is an indicator of a country's economic health, reflecting the performance of companies and the overall business environment. The prices of stocks are determined by supply and demand. Investing in the stock market can be risky, but it can offer the potential for significant returns over the long term. Artificial intelligence, including the stock market, has become increasingly prevalent in the financial sector. Long Short-Term Memory (LSTM) is a type of artificial neural network that is often used in time series analysis. It can effectively predict stock market prices by handling data with multiple input and output timesteps. Metaheuristic algorithms, such as Artificial Rabbits Optimization algorithm (ARO), can be used to optimize the hyperparameters of an LSTM model and improve the accuracy of stock market predictions. In this paper, an optimized deep LSTM network with the ARO model (LSTM-ARO) is created to predict stock prices. DJIA index stocks are used as the dataset. LSTM-ARO is compared with one artificial neural network (ANN) model, three different LSTM models, and LSTM optimized by Genetic Algorithm (GA) model. All the models are tested on MSE, MAE, MAPE, and R2 evaluation criteria. The results show that LSTM-ARO overcomes the other models.

1. Introduction

A stock market is a place where people can buy and sell stocks of companies that are publicly traded with the goal of making money. It is an important indicator of a country's economic health because it reflects the performance of companies and the overall business environment. The stock market operates through a network of exchanges, like Dow Jones (DJIA), which is one of the stock markets in the USA (Gürbüz & Şahbaz, 2022). These exchanges are where buyers and sellers meet to trade stocks. The prices of stocks are determined by the law of supply and demand, where buyers are willing to pay more for a stock if they think it will go up in value, and sellers are willing to accept less if they think it will go down in value (Bosworth et al., 1975; Shah et al., 2019; Teweles & Bradley, 1998).

There are different types of stocks, and they include common, preferred, and warrants. Common stock allows shareholders to own part of a company, vote on company matters, and receive dividends. On the other hand, preferred stock gives shareholders an advantage over common stockholders in terms of receiving dividends and liquidating assets. Lastly, warrants are like options that enable investors to buy a specific number of shares at a set price within a particular period (Giudici et al., 2022; Gornall & Strebulaev, 2022).

The stock market can be affected by different factors that can influence the prices of stocks, such as the state of the economy, political stability, and the performance of individual companies. To predict and analyze stock prices, investors can use different tools and strategies, including fundamental analysis that examines a company's financial situation and industry conditions and technical analysis that considers trends and patterns in stock prices and trading volume(Lim & Rokhim, 2020; Nguyen & Nguyen, 2020). Investing in the stock market is not without risk, as the value of stocks can change a lot. Nevertheless, investing in the stock market can lead to considerable gains over the long term. Mutual funds are a common way for people to invest in the stock market. These funds combine the money of multiple investors and invest in a varied portfolio of stocks. Alternatively, individual investors can invest in stocks directly through brokerage firms (Kim et al., 2020).

Algorithmic trading, also known as automated or black box trading, is the use of computer algorithms to make trading decisions in financial markets. This type of trading is becoming more popular in the stock market because it allows traders to make faster and more accurate

* Corresponding author.
E-mail address: b.gulmez@liacs.leidenuniv.nl.

https://doi.org/10.1016/j.eswa.2023.120346
Received 30 January 2023; Received in revised form 19 April 2023; Accepted 30 April 2023
Available online 4 May 2023

decisions than they could with manual trading. Algorithmic trading systems analyze market data in real-time and identify trading opportunities. They execute trades based on predefined rules and make decisions based on various factors, including market trends, news events, and technical indicators. Both individuals and institutions, including hedge funds, can use algorithmic trading to trade various financial instruments, such as stocks, futures, options, and forex. However, algorithmic trading is not without its risks, and traders must be aware of its potential downsides (Li et al., 2019; Park & Lee, 2021; Tao et al., 2021; Théate & Ernst, 2021).

In recent years, artificial intelligence (AI) has become more prevalent in the financial industry, including the stock market. AI algorithms have the capability to analyze vast amounts of data and make predictions or decisions based on that analysis. This can be useful for predicting stock prices, identifying trends, and making investment decisions. For example, AI algorithms can analyze data like historical stock prices, company financial statements, and market trends in the stock market to predict future performance. AI can also monitor market conditions in real time and identify opportunities for buying or selling stocks. All in all, AI has the potential to significantly improve the efficiency and accuracy of stock market analysis and decision-making, which can ultimately lead to better investment outcomes for investors (Ashta & Herrmann, 2021; Milana & Ashta, 2021).

Long Short-Term Memory (LSTM) is a type of computer program that is designed to analyze and understand data that has long-term dependencies. It is often used in analyzing time series data, such as stock market prices. Time series data refers to the sequence of past stock prices and other financial data, and LSTM can use this data to identify patterns and trends that can be used to predict future stock prices. LSTM is particularly useful in analyzing stock market data because it can handle data with multiple input and output timesteps. For example, a company's stock price may be influenced by various factors such as economic indicators, market trends, and company-specific news. These factors may have a direct or indirect impact on the stock price, and the LSTM model is able to capture these relationships and use them to make more accurate predictions. Overall, LSTM and time series analysis can be a powerful tools for investors who want to make informed investment decisions based on the analysis of stock market data (Gülmez, 2022b; Hu et al., 2021; Mehtab & Sen, 2020).

Metaheuristic algorithms are optimization techniques that can be used to optimize the hyperparameters of an LSTM model. These algorithms are beneficial because they can search for optimal hyperparameters more efficiently and effectively than traditional optimization methods. One benefit of using metaheuristic algorithms is that they can explore a larger search space and find the optimal global solution. Traditional optimization methods may get stuck in a local minimum and not find the best solution. In contrast, metaheuristic algorithms can escape from local minima and find the best solution across the entire search space. Another benefit of metaheuristic algorithms is their ability to handle complex and noisy data. LSTM models can have many hyperparameters, and finding the optimal values for these hyperparameters can be difficult due to the complex nature of the data. However, Metaheuristic algorithms are able to handle this complexity and noise and find the optimal hyperparameters even in these challenging situations (Gülmez, 2023b,a; Kumar & Haider, 2021; Öztürk, 2022).

ARO is a metaheuristic algorithm that is inspired by the behaviour of the rabbits in the nature. It is used to find approximate solutions to optimization and search problems. The basic idea is to use techniques inspired by survival strategies of rabbits, such as foraging, hiding, to generate new solutions to a problem. The solutions, which are often represented as strings of numbers or characters, are evolved over many generations through the application of these algorithm operators until an acceptable solution is found. ARO is a novel metaheuristic algorithm and can be used in a wide range of fields, including artificial intelligence, machine learning, and engineering (Gülmez, 2023a; Wang et al., 2022).

ARO algorithm can be used to optimize the hyperparameters of LSTM networks. Optimizing the hyperparameters of an LSTM network can be challenging, as many different parameters can be adjusted, including the number of layers, the number of neurons per layer, and the learning rate. However, a ARO algorithm can be used to optimize these hyperparameters by treating them as a set of variables in a search space.

2. Literature review

In the literature, there are lot of research on predicting stock prices. Some of them are listed below.

Xie et al. (2021) attempted to solve the issue of typical offline neuro-fuzzy systems needing training data to fully reflect all system behaviors, which may be challenging owing to dramatic swings in data distributions. The authors developed a unique technique that combines a neuro-fuzzy system with the Hammerstein-Wiener model, establishing an indivisible five-layer network. The efficacy of the suggested model was examined on three financial stock datasets and found to outperform existing neuro-fuzzy systems and the standard Hammerstein-Wiener model.

Li et al. (2020) investigated the predictability of investor sentiment in the Chinese stock market by utilizing online user-generated material to gauge sentiment and by comparing various text classification algorithms, price forecasting models, time horizons, and information update systems. The findings demonstrated that daily investor sentiment only provided predicted information for opening prices, but hourly investor sentiment provided two hours of predictive information for closing prices. This indicates that investors adjust their expectations during market hours.

Chen et al. (2019) aimed to develop a stock market trend prediction model using an anticipatory computing approach and public mood and emotion data. The model was tested on a real-world stock market dataset and found to accurately predict stock market trends.

Adhikari et al. (2021) proposed a hybrid deep learning model for stock price prediction using sentiment analysis and found that the model was able to improve prediction accuracy compared to traditional machine learning methods. This suggests that the use of deep learning techniques in combination with traditional machine learning approaches can lead to improved performance in stock price prediction.

Rezaei et al. (2021) presented hybrid algorithms, CEEMD-CNN-LSTM and EMD-CNN-LSTM, which employ full ensemble empirical mode decomposition (CEEMD) and empirical mode decomposition (EMD) to extract deep features and time sequences for one-step-ahead stock price prediction. The results demonstrated that the combination of CNN, LSTM, and CEEMD or EMD improved prediction accuracy and beat other models.

Hammoudeh et al. (2021) used the nonparametric causality-in-quantiles test to investigate the causal connections between oil prices and five clean energy stock indexes. During normal market circumstances, oil returns produced renewable stock index returns, but this was not the case during difficult market situations. Lower quantiles of the volatility study revealed the considerable bidirectional correlation between oil price volatility and renewable energy stock volatility. There were no substantial causal correlations between oil prices and renewable energy stocks during the COVID-19 epidemic.

Maguluri and Ragupathy (2020) used the Auto-Regressive Integrated Moving Average (ARIMA) model to forecast stock prices. Results showed that the model was effective in predicting future prices and forecasting the stock market.

3. Material and method

3.1. Time series analysis

Time series analysis is a statistical method used to analyze data that is collected over a period of time. It involves examining the patterns and

trends in the data to make predictions about future events or to understand the underlying processes that are driving the data. Time series analysis can be used to forecast demand for a product, predict the stock market, or understand the trends in economic indicators such as unemployment or inflation. Time series models use past data to forecast future values and can be used to understand the underlying trends, seasonality, and noise in the data. Time series analysis can also be used to identify and forecast the impact of exogenous variables, such as changes in government policies or technological innovations, on the data. There are several factors that can impact a time series (Chatfield, 2003; Hamilton, 2020; Gülmez, 2021):

Trend: A trend refers to the overall direction of the time series, whether it is increasing, decreasing, or stable over time.

Seasonality: Seasonality refers to the presence of repeating patterns in the data, such as spikes in sales during the holiday season.

Cyclicality: Cyclicality refers to the presence of longer-term patterns in the data, such as economic booms and busts.

Irregularity: Irregularity refers to random or unpredictable fluctuations in the data.

Exogenous factors: Exogenous factors are external factors that can affect the time series, such as economic policies, natural disasters, or social trends.

3.2. Ann

Artificial neural networks (ANNs) are computational models inspired by the structure and function of the human brain. They consist of layers of interconnected nodes or "neurons" that process and transmit information. Each neuron receives input from other neurons and applies a mathematical function to this input to produce an output. The output is then passed on to other neurons in the next layer (Gülmez & Kulluk, 2019).

The structure and function of ANNs can be adjusted through a process called training. During training, the network is presented with a set of inputs and their corresponding outputs, and the network adjusts its weights and biases to minimize the difference between its predicted output and the actual output. This process allows the network to "learn" and improve its performance on tasks such as classification, regression, and clustering (Askarzadeh & Rezazadeh, 2013).

A neuron, also known as a perceptron or processing element, is the fundamental unit of computation in an artificial neural network. It receives input from other neurons or external sources, processes the input using a set of weights and biases, and produces an output based on that processing. The processing occurs through the application of an activation function, which determines the output of the neuron based on the input and weights. Neurons are connected to each other in layers, with the input layer receiving external data and the output layer producing the final result of the network. The intermediate layers, known as hidden layers, process the data and pass it along to the next layer. Neurons within a layer are connected to every neuron in the next layer, allowing the network to learn complex relationships and patterns in the data (Gülcü, 2022).

There are several elements that make up a neuron in an ANN. These include (Krogh, 2008):

Inputs: These are the data that are fed into the neuron, which are typically numerical values.

Weights: These are the values assigned to each input, which are used to determine the importance of that input in the overall calculation of the neuron.

Bias: This is a constant value that is added to the weighted sum of the inputs.

Activation function: This is a mathematical function that determines the output of the neuron based on the inputs and weights. The most common activation functions used in ANNs are sigmoid, tanh, and ReLU.

Output: This is the final result of the neuron's calculation, which is then passed on to other neurons in the network.

Error: This is the difference between the predicted output of the neuron and the actual output, which is used to adjust the weights and biases in order to improve the accuracy of the network.

A layer in an ANN is a group of interconnected neurons that work together to process and transmit information. There are typically multiple layers in an ANN, with the input layer receiving the raw data and the output layer producing the final result. Between the input and output layers are hidden layers, which are responsible for performing the necessary computations and transformations on the input data to produce the desired output. The number of hidden layers and the number of neurons in each layer can vary depending on the specific task the ANN is designed to perform. A layer can be seen in Fig. 1.

3.3. Lstm

LSTM is a type of artificial neural network designed to process sequential data, such as time series, audio, or text. It is particularly useful for processing data with long-term dependencies, where the output at a given time step depends on information from previous time steps. LSTM networks are able to remember this information over a longer period of time by using memory cells, input gates, output gates, and forget gates. These gates control the flow of information into and out of the memory cells, allowing the network to selectively store and retrieve information as needed. LSTMs are commonly used for tasks such as language translation, speech recognition, and stock price prediction. The elements of an LSTM network include (Gers et al., 2000; Yu et al., 2019):

Input gate: Controls the flow of data into the memory cell.

$$inputgate = \sigma(Wi^*[ht-1, xt] + bi) \tag{1}$$

Forget gate: Controls the flow of data out of the memory cell.

$$forgetgate = \sigma(Wf^*[ht-1, xt] + bf) \tag{2}$$

Output gate: Controls the output of the memory cell to the rest of the network.

$$outputgate = \sigma(Wo^*[ht-1, xt] + bo) \tag{3}$$

Cell state: Stores the information in the memory cell.

$$memorycell = ft^*ct-1 + it^*tanh(Wc^*[ht-1, xt] + bc) \tag{4}$$

Hidden state: Output of the LSTM unit, used to make predictions or pass information to the next LSTM unit.

$$hiddenstate = ot^*tanh(ct) \tag{5}$$

3.4. Genetic algorithm (GA)

GA is a computational method that mimics the process of natural selection to find solutions to optimization and search problems. The idea behind it is to use a population of candidate solutions, called individuals, and repeatedly apply genetic operators such as selection, crossover (recombination), and mutation to generate new individuals. The new individuals are then evaluated using a fitness function that measures the quality of the solution. This process is repeated over multiple generations until an acceptable solution is found (Gülmez, 2023a; Gülmez & Korhan, 2021).

GA has three essential components (Alkafaween et al., 2021):

Encoding: Each individual is encoded as a string of numbers or characters, called a chromosome. The encoding is chosen based on the specific problem being solved.

Evaluation: The fitness function is used to evaluate the quality of the solution represented by each individual. The fitness function is designed based on the problem at hand.

Evolutionary Operators: These operators are used to create new individuals from existing ones. The most commonly used operators are selection, crossover, and mutation. Selection is used to pick the best

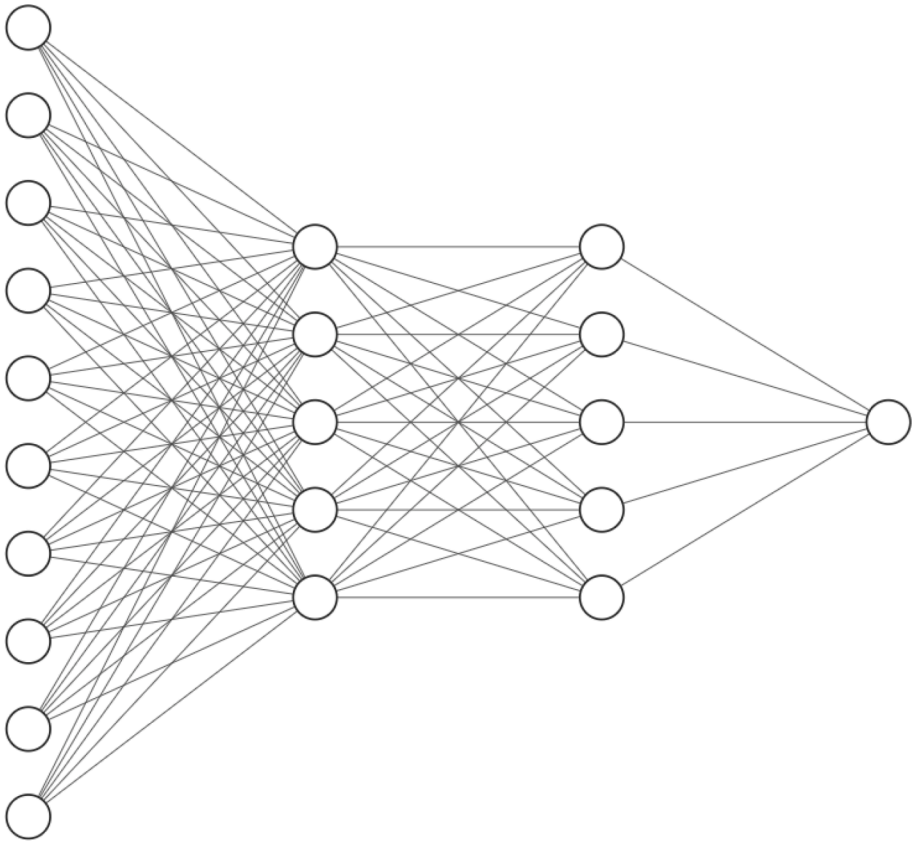


Fig. 1. A sample ANN (Gülmez, 2022b).

individuals to reproduce. Crossover is used to merge the chromosomes of two individuals to form a new individual. The mutation is used to introduce small random changes into the chromosomes of individuals.

It is worth noting that GA is a heuristic optimization method, and it is not guaranteed to find the optimal global solution, but it can provide a good solution with reasonable computational cost. However, it may be computationally intensive and time-consuming for large-scale problems, especially if the dataset is large and the training process is prolonged (Gülmez, 2022c).

3.5. Artificial rabbits optimization (ARO)

ARO is a novel algorithm created by Wang et al. (Wang et al., 2022). Rabbits are plant-eaters, feeding mainly on grass, forbs, and leafy weeds. They have evolved various survival strategies over time. One of their strategies is to avoid eating the grass close to their burrows to prevent predators from finding their nests. Rabbits have an extensive field of vision, with most of it devoted to overhead scanning, making it easier for them to find food over a large area. This strategy of foraging away from their nests is known as exploration and is reflected in the Chinese idiom "rabbits do not eat the grass near its own nest." Another strategy is random hiding. Rabbits are skilled at creating burrows to evade predators or hunters. They dig multiple burrows around their nest and select one randomly as a shelter from predators. With short forelegs and long back legs, rabbits can run quickly due to strong muscles and tendons. They can also abruptly stop while running, turn sharply, or even run back in a zigzag motion to confuse their enemies and increase their chances of survival. This strategy is called exploitation. Since rabbits are at the bottom of the food chain and have many predators, they must conserve energy to survive. To do this, they adaptively switch between detour foraging and random hiding based on their energy levels (Gülmez, 2022a; Wang, Cao, Zhang, Mirjalili, & Zhao, 2022).

ARO has two stages. They are detour foraging (exploration) and random hiding (exploitation).

When rabbits forage, they often go far away and ignore nearby grass. This behavior is called detour foraging. In ARO, each rabbit has its own region with grass and burrows. They randomly visit each other's regions to forage and may perturb around a food source. This means they update their position towards another rabbit in the swarm and add a perturbation. A model has been proposed to describe this behavior.

$$\overrightarrow{v}_i(t+1) = \overrightarrow{x}_j(t) + R\cdot(\overrightarrow{x}_i(t) - \overrightarrow{x}_j(t)) + round(0.5\cdot(0.05 + r_1))\cdot n_1, \tag{6}$$

$$R = L\cdot c \tag{7}$$

$$L = (e - e^{(\frac{t-1}{T})^2})\cdot sin(2\pi r_2) \tag{8}$$

$$c(k) = \begin{cases} 1 & if k == g(l) \\ 0 & else \end{cases} k = 1, \cdots, d \text{ and} l = 1, \cdots, \lceil r_3\cdot d \rceil \tag{9}$$

$$g = randperm(d) \tag{10}$$

$$n_1 \sim N(0,1) \tag{11}$$

The candidate position of each rabbit at time $t+1$ is $v_i(t+1)$, while $x_i(t)$ is its position at time t. Other variables include n (rabbit population size), d (problem dimension), and T (maximal number of iterations). Perturbation and running length (L) aid in global search, with longer step sizes promoting exploration and shorter ones promoting exploitation. The mapping vector c helps randomly select search individuals to mutate. The running operator (R) simulates rabbit running behavior, probing the search space with sudden turns in random directions. Detour foraging enables rabbits to move to other regions, promoting the

exploration and the global search capability of the algorithm. Eq. (6) perturbation helps avoid local extrema and perform the global search, while Eq. (8) generates longer steps during initial iterations and shorter steps later. Mapping vector c selects individuals for mutation during foraging. The running operator R simulates rabbit movements. Eq. (6) shows that search individuals explore by randomly searching for food based on the position of other rabbits. This unique behavior allows them to venture far from their own territory and discover new regions, ensuring global search capability in the ARO algorithm.

ARO uses random hiding to exploit the search space and avoid predators. Each rabbit generates d burrows around its nest in each dimension of the search space and chooses one at random to hide in. This helps reduce the risk of being preyed upon. The jth burrow of the ith rabbit is generated using the following equation:

$$\vec{b}_{ij}(t) = \vec{x}_i(t) + H \cdot g \cdot \vec{x}_i(t), i = 1, \cdots, n \text{ and } j = 1, \cdots, d \tag{12}$$

$$H = \frac{T - t + 1}{T} \cdot r_4 \tag{13}$$

$$n_2 \sim N(0, 1) \tag{14}$$

$$g(k) = \begin{cases} 1 & if k == j \\ 0 & else \end{cases} k = 1, \cdots, d \tag{15}$$

Rabbits create d burrows around themselves along each dimension of the search space to avoid predators. Eq. (12) shows how these burrows are generated. The hiding parameter, H, decreases linearly from 1 to 1/T over iterations, with a random perturbation. This causes the burrows to be generated in a larger neighborhood initially and gradually decrease as iterations progress.

$$\vec{v}_i(t+1) = \vec{x}_i(t) + R \cdot (r_4 \cdot \vec{b}_{i,r}(t) - \vec{x}_i(t)), i = 1, \cdots, n \tag{16}$$

$$g_r(k) = \begin{cases} 1 & if k == \lceil r_5 \cdot d \rceil \\ 0 & else \end{cases} k = 1, \cdots, d \tag{17}$$

$$\vec{b}_{i,r}(t) = \vec{x}_i(t) + H \cdot g_r \cdot \vec{x}_i(t) \tag{18}$$

A rabbit selects one of its d burrows for hiding using a random number b_{ir}, and two random numbers r_4 and r_5. The ith search individual updates its position towards this selected burrow according to Eq. (16).

The ith rabbit's position is updated after either detour foraging or random hiding as follows:

$$\vec{x}_i(t+1) = \begin{cases} \vec{x}_i(t) & f(\vec{x}_i(t)) \le f(\vec{v}_i(t+1)) \\ \vec{v}_i(t+1) & f(\vec{x}_i(t)) > f(\vec{v}_i(t+1)) \end{cases} \tag{19}$$

Eq. (19) means that if the fitness of the candidate position of the ith rabbit is superior to its current position, it will move to the candidate position generated by either Eq. (6) or Eq. (16).

ARO rabbits switch from detour foraging to random hiding as the iterations progress, which is based on their energy level. The energy factor in ARO represents this switch from exploration to exploitation and is defined as Eq. (20).

$$A(t) = 4\left(1 - \frac{t}{T}\right) ln \frac{1}{r} \tag{20}$$

3.6. Models

In this study, one ANN and five LSTM model are used. The LSTM models are LSTM1D, LSTM2D, LSTM3D, LSTM-GA, and the last and proposed model, the LSTM model optimized by ARO (LSTM-ARO).

3.6.1. Ann

Table 1 shows ANN architecture. The input layer has 20 neurons, and this is the input dimension of the network. The first dense layer has 10 neurons, this layer is fully connected to the input layer. The second dense layer also has 10 neurons, this layer is fully connected to the first dense layer. The dropout layer has a dropout rate of 0.5, this layer is applied between the second dense and the output layer. The output layer has 1 neuron, this is the output dimension of the network. It is fully connected to the dropout layer. This network takes an input of size 20 and passes it through two dense layers, each with 10 neurons. In between the two dense layers, there is a dropout layer with a dropout rate of 0.5 which will randomly drop out 50% of the neurons from the second dense layer during training. Finally, the output of the dropout layer is passed through the output layer which has 1 neuron and produces the output of the network.

Table 1
ANN model architecture.

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

3.6.2. Lstm1d

Table 2 shows LSTM1D architecture. The first layer is the input layer, which has 20 nodes. The second layer is an LSTM layer, which has 10 nodes. The third layer is a dense layer, which also has 10 nodes. The fourth layer is a dropout layer, which has a dropout rate of 0.5. The final layer is a dense layer with 1 node, which is an output layer.

3.6.3. Lstm2d

Table 3 shows LSTM2D architecture. It consists of four layers: an input layer, two LSTM layers, a dense layer, and a final dense layer with a dropout rate of 0.5. The input layer has 20 neurons, the first and second LSTM layers have 10 neurons each, the dense layer has 10 neurons, and the final dense layer has 1 neuron. This architecture utilizes the sequential nature of LSTM layers to process input data, followed by dense layers for further feature extraction and dropout for regularization. The final dense layer with 1 neuron is used for output.

3.6.4. Lstm3d

Table 4 shows LSTM3D architecture. The architecture of the model consists of an input layer with 20 inputs, followed by three LSTM layers with 10 units each, a dense layer with 10 units, a dropout layer with a dropout rate of 0.5, and finally a dense layer with 1 unit as the output layer.

3.6.5. LSTM-GA and LSTM-ARO

Table 5 shows LSTM-GA and LSTM-ARO architectures. This neural network architecture consists of several layers, including an input layer with 20 neurons, and three LSTM layers with x_0, x_2 and x_4 neurons, respectively. If the values of x_1 and x_3 is 0, these layers do not exist. The second and third layer is connected to variables. A dense layer with x_5 neurons, a dropout layer with a dropout rate of x_6, and an output layer with 1 neuron. X_0, x_2, x_4, x_5, x_6 values are variables from GA and ARO.

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

Table 5
LSTM-GA and LSTM-ARO 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

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

Also optimizer and learning rate are other variables. GA and ARO find the best variable values for the architecture.

Hyperparameter alternatives of the LSTM-GA and LSTM-ARO models are seen in Table 6. GA and ARO search and try to find best hyperparameters.

LSTM-ARO model can be seen in Fig. 2.

4. Results and discussion

4.1. Dataset

DJIA is the abbreviation for Dow Jones,. It is the market for trading equities, bonds, and other securities in USA.

Yahoo Finance can be used to collect DJIA data from the internet. It is a financial news and data website that provides various financial data, including stock prices, market indices, exchange rates, and more. Python is used with Yahoo Finance API to access and download DJIA data for specific time periods (Garita, 2021; Yahoo Finance, 2023).

Price prediction has been made with time window sized 20 previous days. The input is prices of 20 days. The output is the price of the next day. The dataset time range is between 2018.01.01 and 2023.01.01. It is 5 years stock market data. 80 percent of the data is used for training, 20 percent of the data is used for test.

4.2. Evaluation criteria

Evaluation criteria for regression models are used to measure the accuracy and effectiveness of the model in predicting the output values based on the input data (Gülmez, 2022a).

MSE is a measure of the difference between the predicted values and the actual values. It is calculated by taking the difference between the predicted and actual values and squaring them, then taking the average of all these squared differences. This value is used to assess the accuracy of the model, and the lower the MSE, the better the model.

$$MSE = \frac{1}{n}\sum_{i}^{n}(y_i - \hat{y}_i)^2 \tag{21}$$

MAE is another measure of the difference between the predicted and actual values. It is calculated by taking the absolute value of the difference between the predicted and actual values and taking the average of all these differences. This value is also used to assess the accuracy of the model and the lower the MAE, the better the model.

$$MAE = \frac{1}{n}\sum_{1}^{n}|y_i - \hat{y}_i| \tag{22}$$

MAPE is a measure of the accuracy of a model that is expressed as a percentage. It is calculated by taking the absolute value of the difference between the predicted and actual values and dividing it by the actual value, then taking the average of all these percentages. This value is used to assess the accuracy of the model and the lower the MAPE, the better the model.

$$MAPE = \frac{100}{n}\sum_{i}^{n}\frac{y_i - \hat{y}_i}{y_i} \tag{23}$$

R2 score is a measure of how well a model fits the data. It is calculated by taking the sum of the squares of the differences between the predicted and actual values, dividing it by the sum of the squares of the differences between the mean of the actual values and the actual values, and then subtracting this value from 1. This value is used to assess the accuracy of the model and the closer it is to 1, the better the model.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \tag{24}$$

4.3. Results

In this study, various algorithms, including LSTM1D, LSTM2D, LSTM3D, ANN, LSTM-GA, and optimized LSTM with ARO (LSTM-ARO) are tested on the DJIA stock dataset from 2018 to 2023. The results of these algorithms are then carefully analyzed and recorded. The aim is to determine which algorithm performs best in terms of accuracy and efficiency for predicting stock prices. The evaluation criteria used include MSE, MAE, MAPE, and R2 score. The results of the testing are crucial in determining the most effective approach for predicting stock prices in the DJIA market.

The proposed model LSTM-ARO gives successful results. The graphs of the results can be seen Fig. 3. Blue and dotted line is the real values, orange lines are predictions. The figure shows generally predictions are successful. Also, Fig. 4 shows LSTM-GA results, Fig. 5 shows LSTM1D results, Fig. 6 shows LSTM2D results, Fig. 7 shows LSTM3D results, and Fig. 8 shows ANN results.

For MSE evaluation criteria, the models are compared. The comparison and the results are seen in Table 7. The results appear to be a table of mean squared errors (MSE) for various ticker symbols and

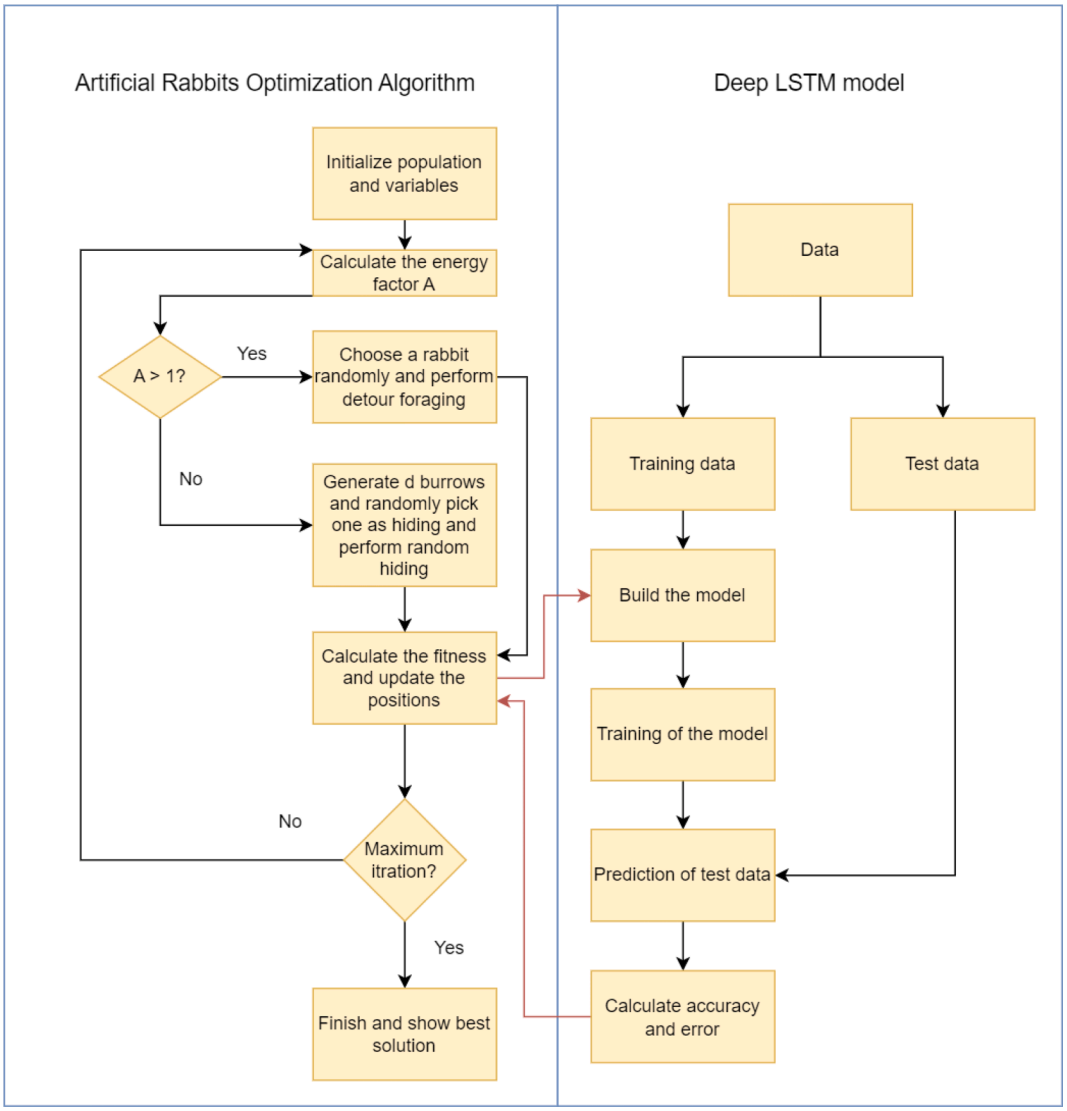


Fig. 2. LSTM-ARO model.

different models. The lower the MSE, the better the model is at predicting the true values. From the table, it can be seen that the LSTM-ARO model has the lowest MSE for most of the ticker symbols, indicating that it may be the best model for predicting the stock prices.

Table 8 shows the MAE results of the models. The MAE is a measure of how accurate the predictions made by each model are, with lower values indicating better performance. It appears that in general, the LSTM-ARO model has the lowest MAE across most of the stocks, indicating that it is the most accurate model. However, it's important to notice that the results vary depending on the stock. Some stocks are better predicted by different models.

Table 9 shows the results of MAPE values for different stock tickers and different models. The MAPE is a measure of the prediction accuracy of a model, where a lower MAPE value indicates a better fit. From the table, it appears that the LSTM-ARO model generally has the lowest MAPE values across all tickers, followed by the ANN model. The LSTM1D, LSTM2D, and LSTM3D models generally have higher MAPE values, indicating lower prediction accuracy.

Table 10 shows the results of R2 score for each stock ticker, using different models. R2 is a measure of how well the model fits the data, where a score of 1.0 means a perfect fit and a score of 0 means the model does not explain any of the variance in the data. From the results, it appears that the LSTM-ARO model generally has the highest R2 scores, although there are some outliers where other models perform better. It is also worth noting that some of the models, such as LSTM1D and LSTM2D, have negative R2 scores for certain tickers, indicating that they are performing worse than a simple average of the data. Overall, it looks like the LSTM-ARO is the best model for fitting the data.

The paper proposes using various deep learning models, such as LSTM-ARO, ANN, LSTM1D, LSTM2D, and LSTM3D, to predict future stock market prices using the DJIA dataset. The results show that the LSTM-ARO model outperforms the other models in terms of MSE, MAE, MAPE, and R2 scores. This suggests that using ARO algorithm optimization in combination with LSTM is an effective approach for predicting stock market prices in this dataset.

Table 11 shows the time spent predicting DIAJ stocks' future prices. It is in second. ANN is the fastest algorithm. Then LSTM1D, LSTM-GA, LSTM-ARO, LSTM2D, and LSTM3D follow. LSTM-ARO and LSTM-GA algorithms are worked with 50 iterations and 50 population numbers. So, the total time of the response of the algorithms is 754,890 s (8.737 days) and 721,660 s (8.353 days). The total time is used only one time to get the best models. Then in a real dynamic trading system, the models will run one time per day. So, the execution times are about 5 min for the total of the DJIA stocks. It is so enough.

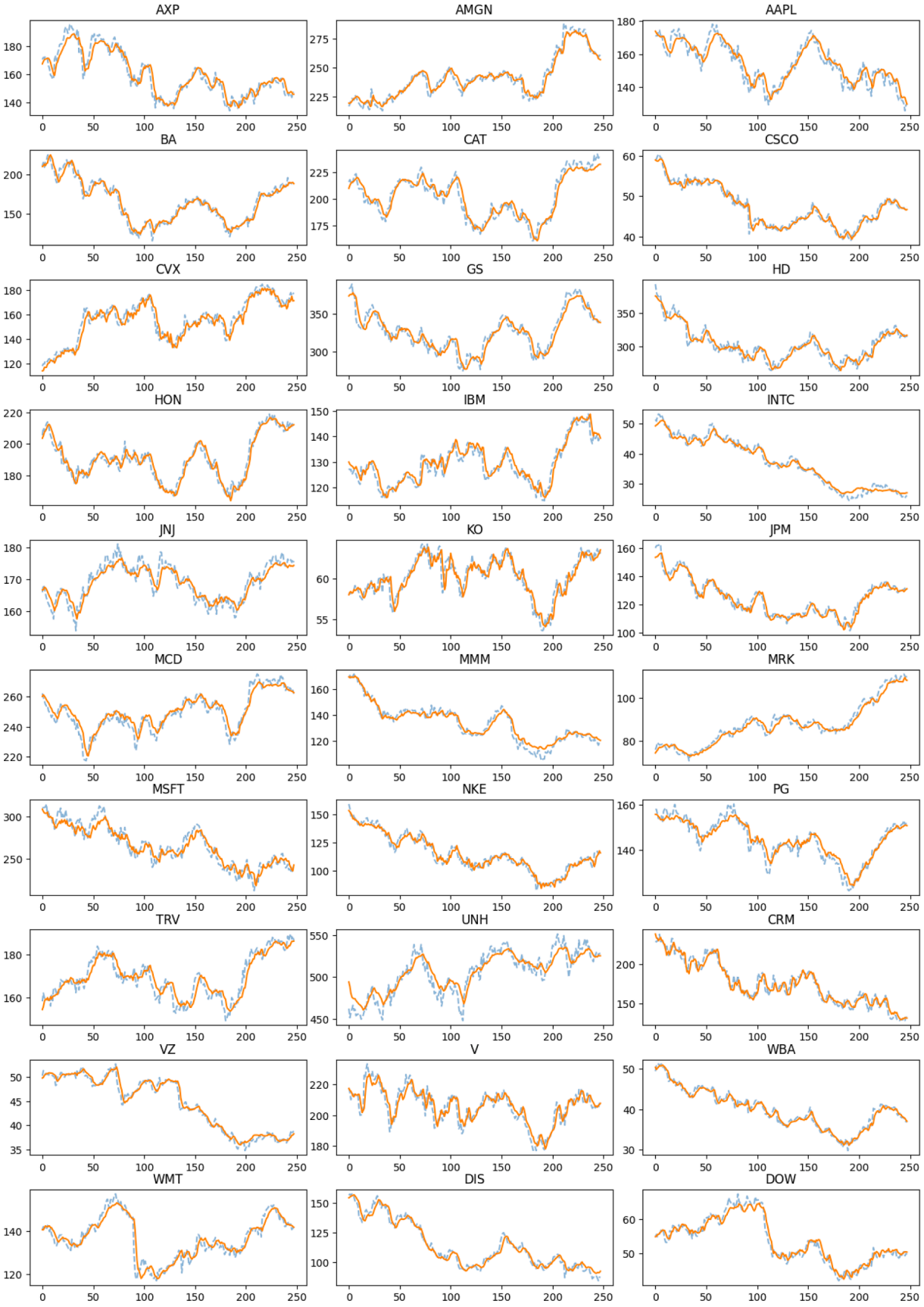


Fig. 3. LSTM-ARO results.

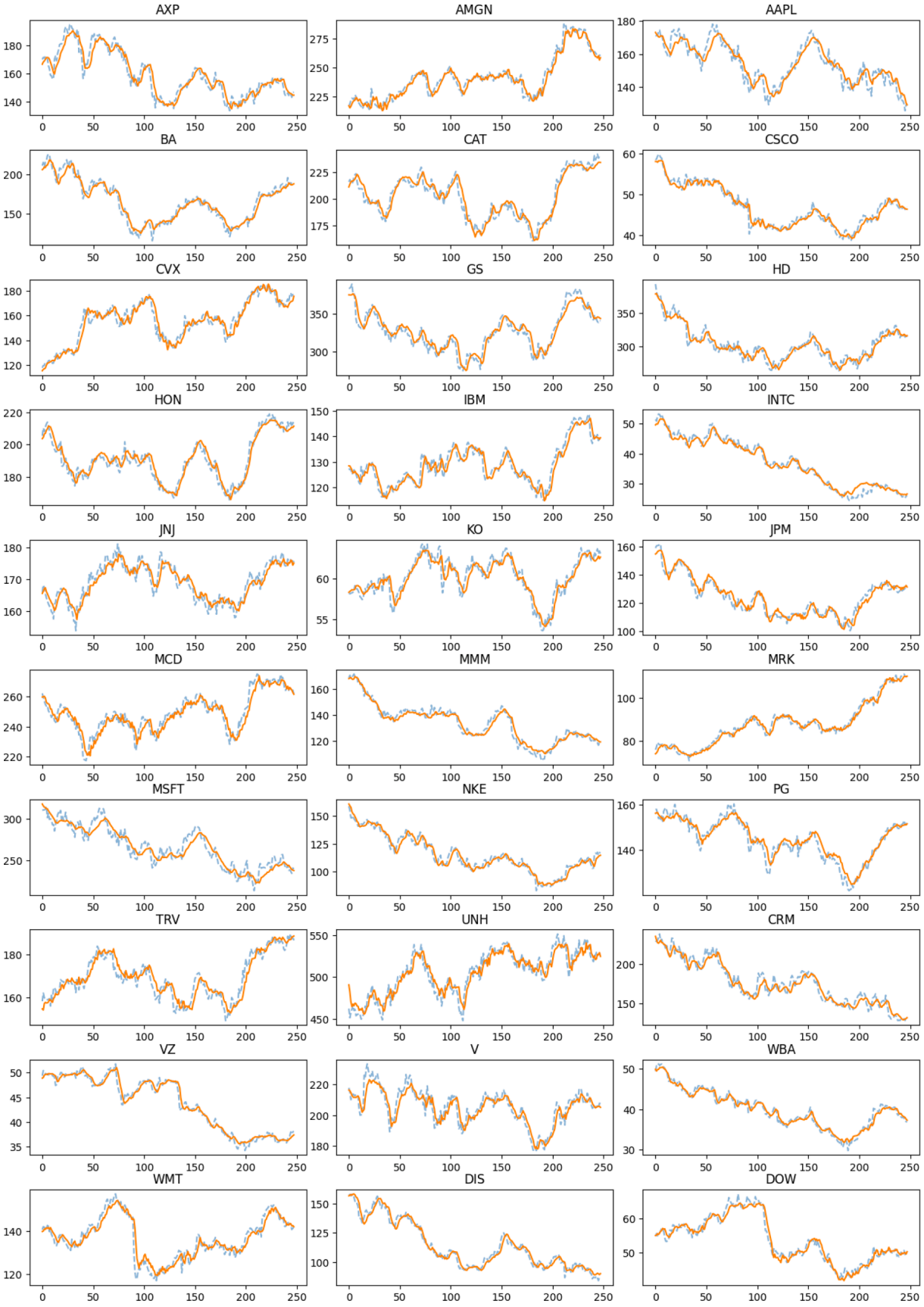


Fig. 4. LSTM-GA results.

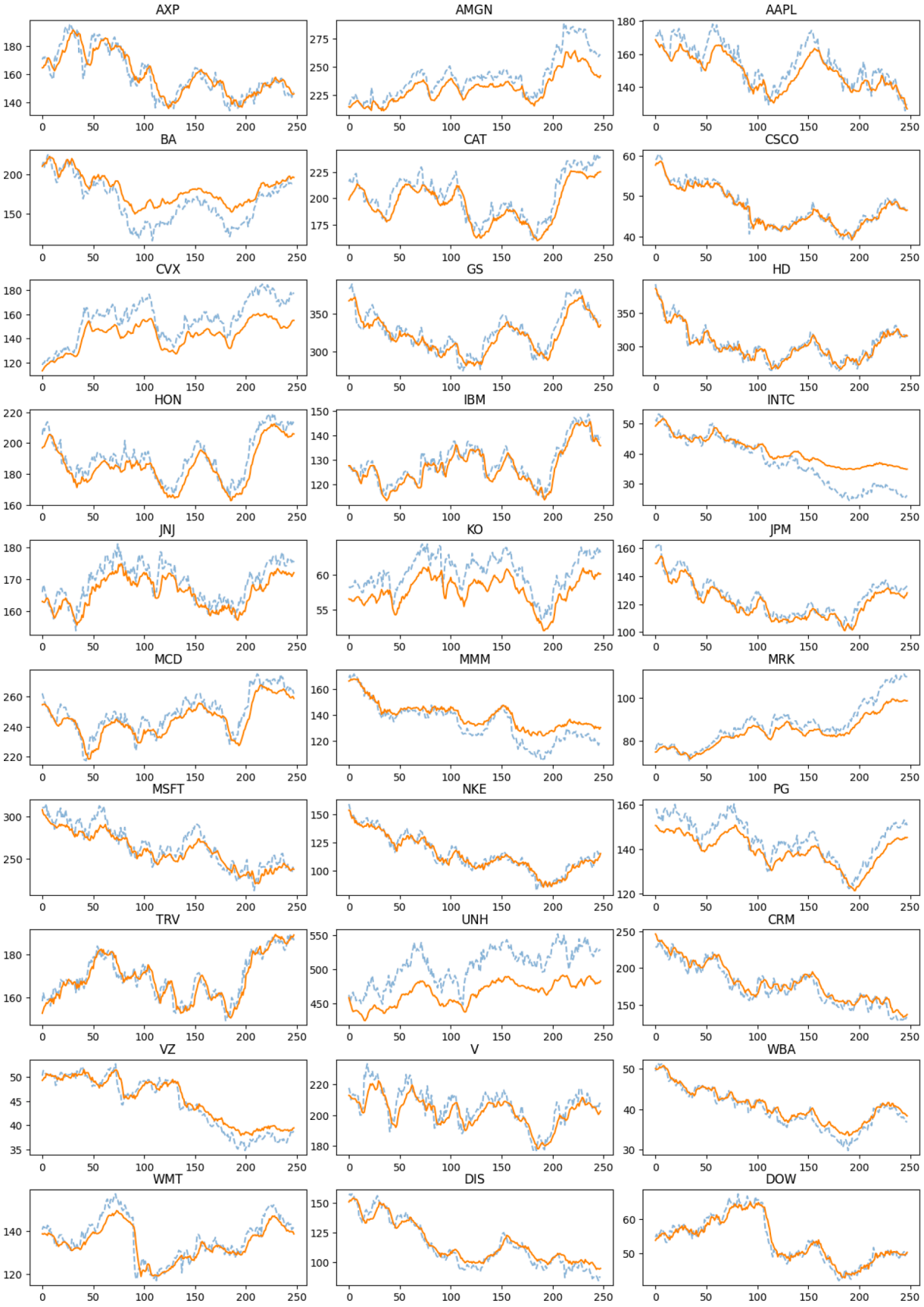


Fig. 5. LSTM1D results.

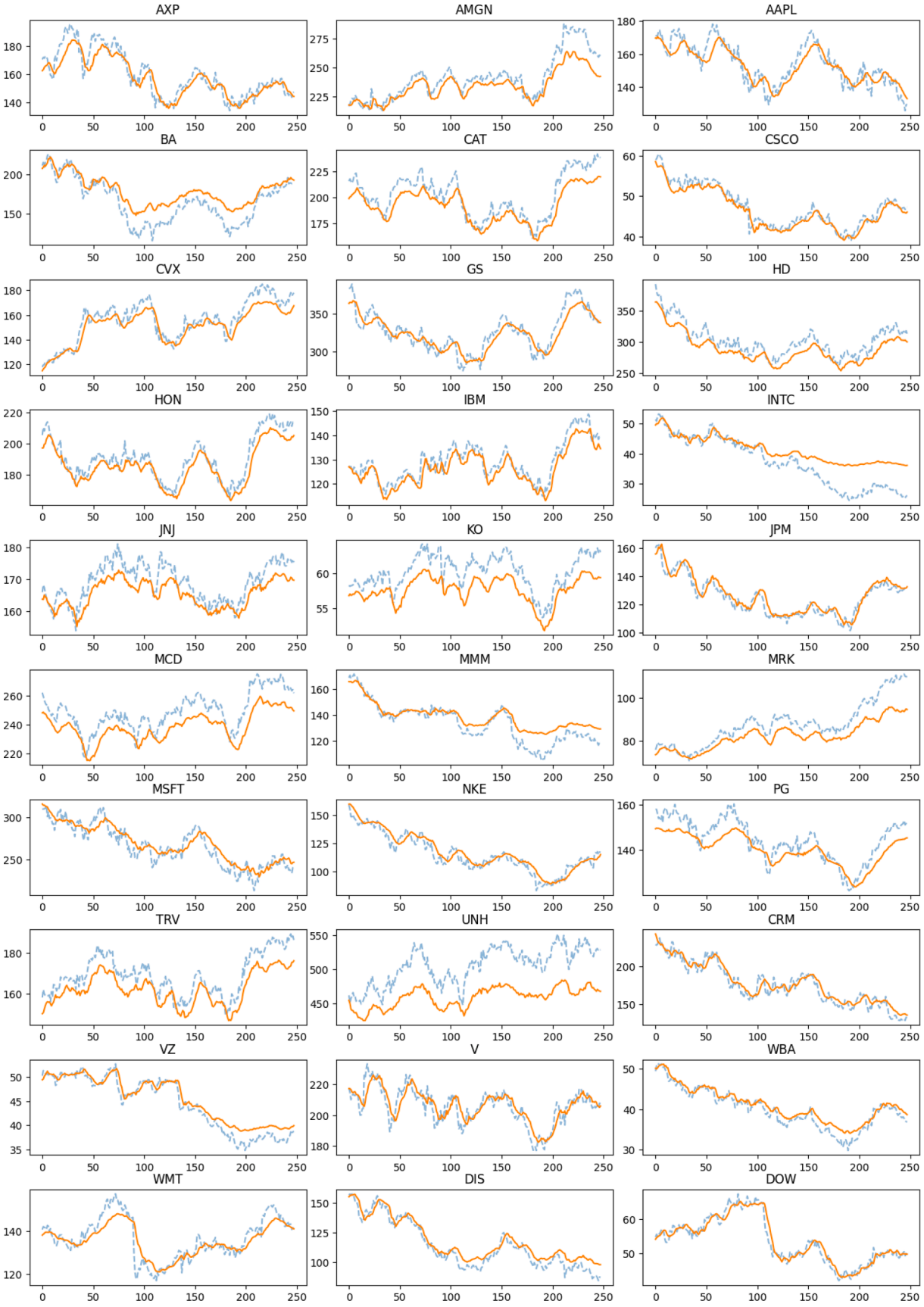


Fig. 6. LSTM2D results.

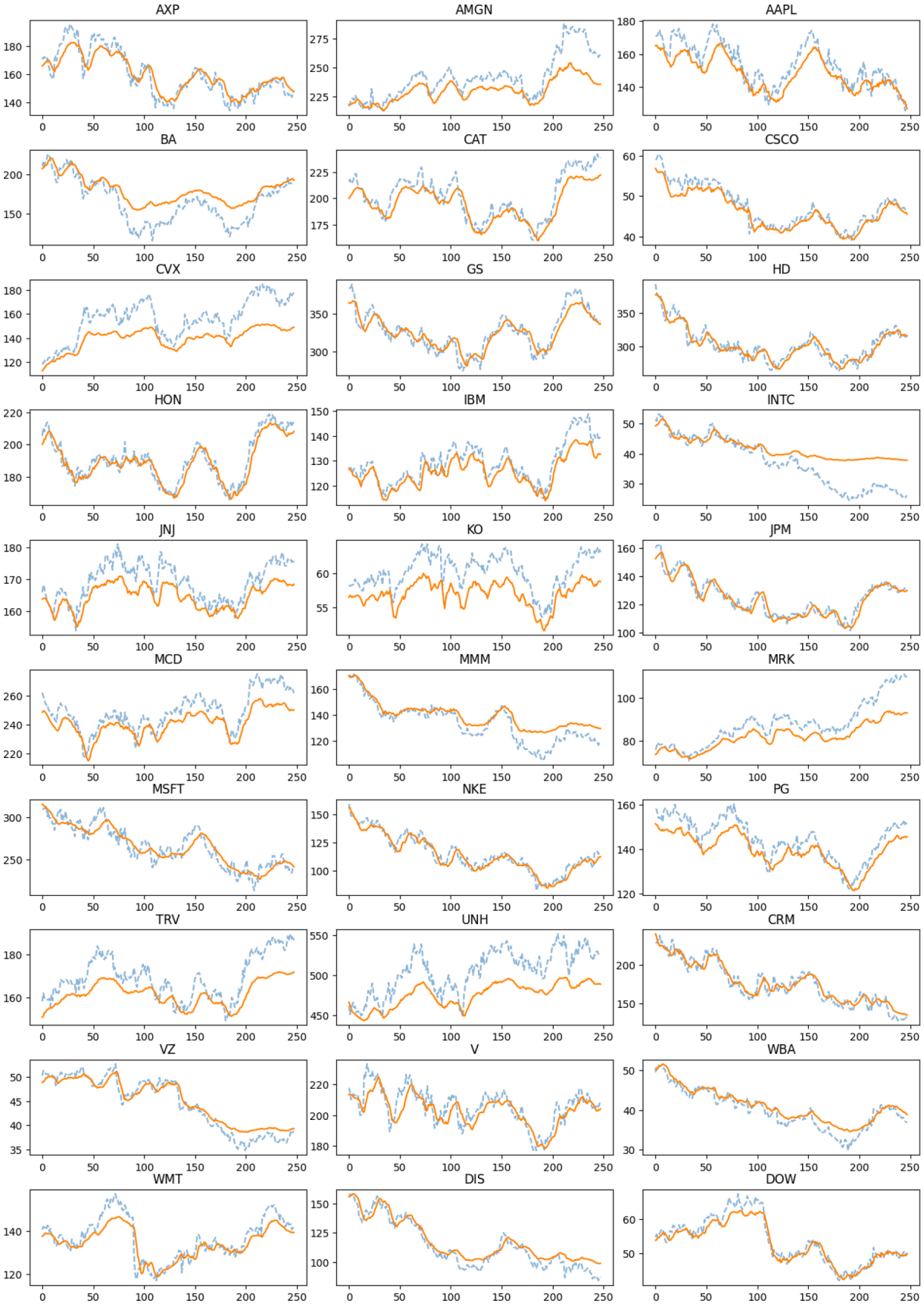


Fig. 7. LSTM3D results.

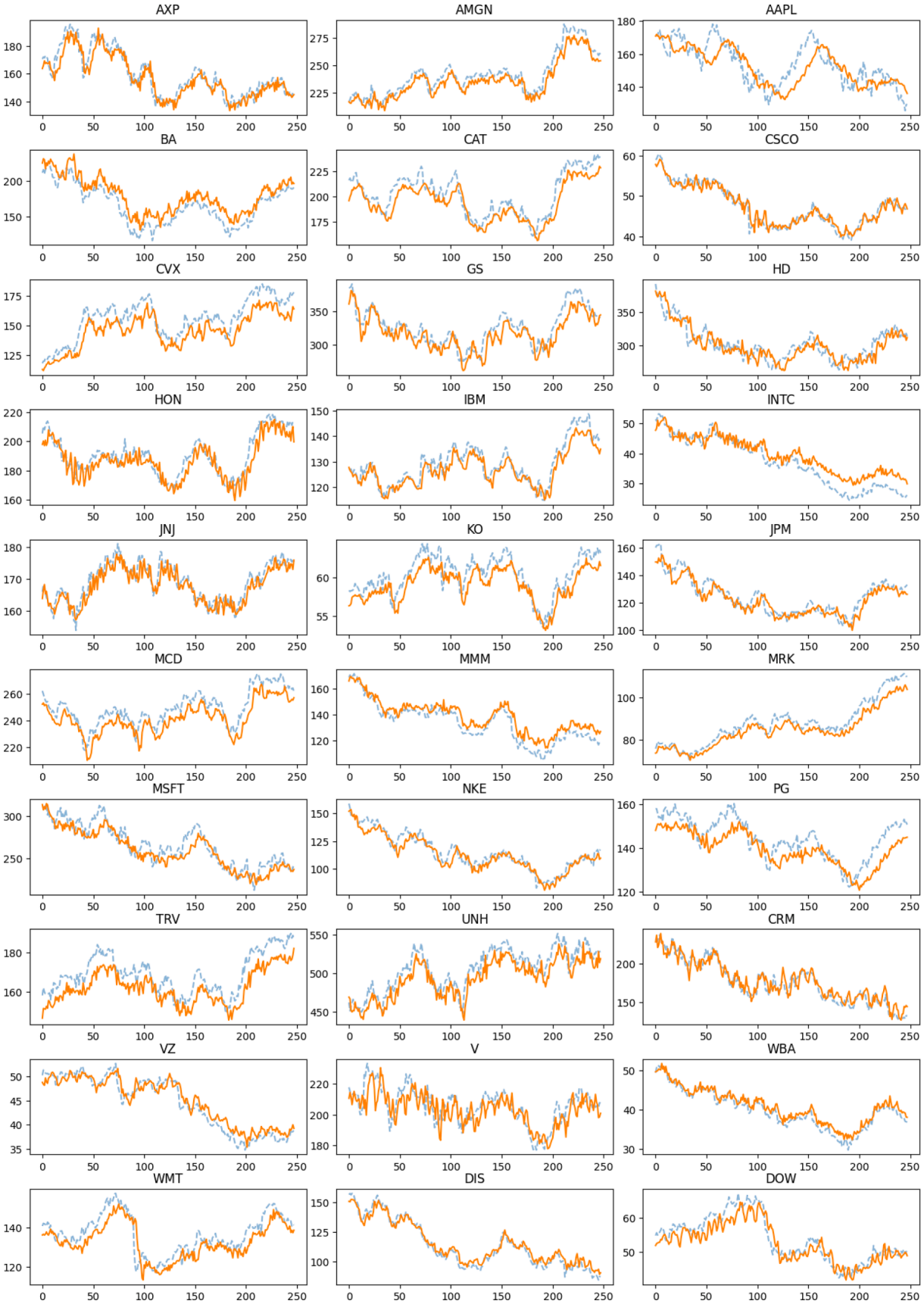


Fig. 8. ANN results.

Table 7
Comparison of the models for MSE criteria.

MSE Ticker	LSTM-ARO	LSTM-GA	LSTM1D	LSTM2D	LSTM3D	ANN
AXP	**24.287**	25.869	42.698	48.079	42.544	35.997
AMGN	**19.774**	20.272	136.640	105.350	224.140	57.672
AAPL	**22.731**	24.109	51.696	37.606	63.224	86.148
BA	**33.658**	40.216	348.538	276.665	339.215	249.795
CAT	38.438	**36.329**	116.124	147.690	112.921	124.205
CSCO	**0.990**	1.143	1.219	2.209	3.303	1.708
CVX	23.970	**21.953**	209.010	53.453	328.273	144.246
GS	**80.137**	80.827	137.054	154.742	123.113	246.217
HD	62.531	59.791	**57.983**	263.135	83.794	185.916
HON	**13.264**	14.589	50.653	43.121	22.091	50.258
IBM	7.333	**7.292**	11.009	14.440	22.188	12.824
INTC	**1.835**	1.990	27.095	33.376	44.033	13.097
JNJ	5.703	**5.023**	16.653	20.351	27.716	7.351
KO	0.957	**0.931**	7.499	8.055	11.691	2.788
JPM	**10.404**	12.926	31.059	19.036	18.637	33.403
MCD	**20.599**	21.524	51.196	142.521	98.483	91.713
MMM	11.472	**10.384**	65.196	59.223	65.675	45.822
MRK	3.690	**2.353**	23.848	46.904	54.171	13.907
MSFT	**63.322**	79.253	119.505	105.814	105.938	133.843
NKE	**14.071**	14.894	17.662	31.001	25.158	34.317
PG	8.289	**8.036**	35.809	34.103	35.374	47.801
TRV	13.135	13.080	12.823	62.723	86.064	67.192
UNH	126.133	**96.373**	1794.141	2308.528	1077.909	239.187
CRM	**39.904**	52.945	81.711	65.323	62.698	74.177
VZ	0.744	**0.728**	2.500	2.703	2.351	2.478
V	**24.406**	25.667	45.692	38.342	51.119	70.378
WBA	**0.749**	0.800	1.999	2.525	2.791	2.079
WMT	**10.827**	11.050	18.483	23.167	22.927	37.360
DIS	**13.959**	14.591	30.200	40.302	41.830	18.364
DOW	**1.905**	2.005	2.408	2.692	3.582	8.339

Table 8
Comparison of the models for MAE criteria.

MAE Ticker	LSTM-ARO	LSTM-GA	LSTM1D	LSTM2D	LSTM3D	ANN
AXP	**3.804**	3.848	5.159	5.415	5.082	4.704
AMGN	**3.318**	3.425	9.614	7.867	11.513	6.193
AAPL	**3.846**	3.955	5.851	4.878	6.632	7.673
BA	**4.425**	4.805	15.994	14.042	15.369	13.813
CAT	4.947	**4.748**	8.875	10.310	8.661	9.235
CSCO	**0.744**	0.813	0.828	1.176	1.430	0.988
CVX	3.830	**3.531**	12.940	5.969	16.006	10.518
GS	**7.060**	7.272	9.653	10.129	8.844	12.525
HD	6.206	**6.026**	**6.046**	14.317	7.487	10.753
HON	**2.876**	3.068	5.851	5.413	3.854	5.769
IBM	2.113	**2.096**	2.652	3.117	3.755	2.836
INTC	1.092	**1.072**	4.107	4.486	5.064	3.059
JNJ	1.912	**1.795**	3.436	3.860	4.525	2.139
KO	**0.738**	0.751	2.584	2.624	3.217	1.439
JPM	**2.523**	2.894	4.570	3.415	3.380	4.569
MCD	**3.630**	3.646	5.864	10.878	8.596	8.233
MMM	2.602	**2.532**	6.410	5.845	6.210	5.538
MRK	1.551	**1.212**	3.810	5.563	5.906	3.165
MSFT	**6.584**	7.583	8.889	8.659	8.784	9.363
NKE	**2.907**	3.050	3.371	4.501	3.939	4.563
PG	2.180	**2.105**	5.392	5.186	5.357	5.968
TRV	2.853	2.808	2.796	7.112	7.962	7.323
UNH	9.016	**7.852**	40.169	45.718	29.796	13.116
CRM	**5.024**	5.879	7.527	6.616	6.419	6.931
VZ	0.618	**0.609**	1.274	1.274	1.258	1.303
V	**3.724**	3.849	5.347	4.752	5.583	6.540
WBA	**0.667**	0.693	1.124	1.270	1.313	1.153
WMT	**2.291**	2.309	3.314	3.464	3.648	4.770
DIS	**2.923**	2.935	4.472	5.300	5.338	3.431
DOW	**1.059**	1.114	1.178	1.203	1.489	2.337

Table 9
Comparison of the models for MAPE criteria.

MAPE Ticker	LSTM-ARO	LSTM-GA	LSTM1D	LSTM2D	LSTM3D	ANN
AXP	**0.024**	0.024	0.032	0.033	0.031	0.029
AMGN	**0.014**	0.014	0.038	0.031	0.045	0.025
AAPL	**0.025**	0.026	0.037	0.032	0.042	0.050
BA	**0.027**	0.030	0.108	0.095	0.105	0.089
CAT	0.025	**0.024**	0.043	0.049	0.042	0.045
CSCO	**0.016**	0.017	0.017	0.025	0.029	0.021
CVX	0.025	**0.023**	0.080	0.037	0.098	0.066
GS	**0.022**	0.022	0.029	0.031	0.027	0.038
HD	0.021	0.020	**0.020**	0.047	0.025	0.035
HON	**0.015**	0.016	0.030	0.028	0.020	0.030
IBM	0.017	**0.016**	0.020	0.024	0.028	0.022
INTC	0.031	**0.031**	0.136	0.150	0.170	0.096
JNJ	0.011	**0.011**	0.020	0.023	0.026	0.013
KO	**0.012**	0.013	0.043	0.043	0.053	0.024
JPM	**0.020**	0.024	0.036	0.027	0.027	0.036
MCD	**0.015**	0.015	0.023	0.043	0.034	0.033
MMM	0.020	**0.019**	0.052	0.048	0.051	0.044
MRK	0.017	**0.014**	0.041	0.060	0.063	0.035
MSFT	**0.025**	0.029	0.033	0.033	0.033	0.035
NKE	**0.026**	0.027	0.030	0.040	0.034	0.039
PG	0.015	**0.015**	0.037	0.035	0.037	0.041
TRV	0.017	0.017	0.017	0.041	0.046	0.043
UNH	0.018	**0.016**	0.079	0.089	0.058	0.026
CRM	**0.029**	0.034	0.044	0.039	0.038	0.041
VZ	0.014	**0.014**	0.031	0.031	0.030	0.030
V	**0.018**	0.019	0.026	0.023	0.027	0.032
WBA	**0.017**	0.018	0.030	0.034	0.035	0.030
WMT	**0.017**	0.017	0.024	0.025	0.026	0.035
DIS	**0.026**	0.026	0.041	0.050	0.050	0.032
DOW	**0.020**	0.021	0.022	0.023	0.027	0.043

Table 10
Comparison of the models for R2 criteria.

R2 score Ticker	LSTM-ARO	LSTM-GA	LSTM1D	LSTM2D	LSTM3D	ANN
AXP	**0.907**	0.900	0.836	0.815	0.836	0.862
AMGN	**0.943**	0.942	0.606	0.696	0.354	0.834
AAPL	**0.857**	0.848	0.675	0.764	0.602	0.458
BA	**0.954**	0.945	0.527	0.625	0.540	0.661
CAT	0.909	**0.914**	0.724	0.649	0.732	0.705
CSCO	**0.961**	0.954	0.952	0.913	0.870	0.933
CVX	0.911	**0.919**	0.227	0.802	−0.214	0.467
GS	**0.886**	0.885	0.806	0.781	0.826	0.651
HD	0.902	0.906	**0.909**	0.587	0.868	0.708
HON	**0.928**	0.921	0.725	0.766	0.880	0.728
IBM	0.884	**0.885**	0.826	0.772	0.650	0.798
INTC	**0.972**	0.970	0.593	0.498	0.338	0.803
JNJ	0.834	**0.854**	0.515	0.407	0.192	0.786
KO	0.841	**0.845**	−0.246	−0.339	−0.943	0.537
JPM	**0.940**	0.924	0.821	0.890	0.892	0.807
MCD	**0.875**	0.869	0.689	0.133	0.401	0.442
MMM	0.947	**0.952**	0.700	0.728	0.698	0.789
MRK	0.962	**0.976**	0.756	0.520	0.446	0.858
MSFT	**0.892**	0.865	0.797	0.820	0.820	0.773
NKE	**0.949**	0.946	0.935	0.887	0.908	0.874
PG	0.900	**0.903**	0.568	0.589	0.574	0.424
TRV	0.866	0.866	0.869	0.359	0.121	0.314
UNH	0.823	**0.865**	−1.517	−2.239	−0.512	0.664
CRM	**0.947**	0.930	0.892	0.913	0.917	0.902
VZ	**0.975**	0.975	0.917	0.910	0.922	0.918
V	**0.814**	0.805	0.652	0.708	0.611	0.465
WBA	**0.967**	0.965	0.913	0.890	0.879	0.910
WMT	**0.888**	0.885	0.808	0.760	0.762	0.612
DIS	**0.965**	0.963	0.924	0.899	0.895	0.954
DOW	**0.956**	0.953	0.944	0.937	0.917	0.806

Table 11
Time spent predicting DJIA stocks.

Model	LSTM-ARO	LSTM-GA	LSTM1D	LSTM2D	LSTM3D	ANN
Time	301.956	288.664	249.576	374.284	501.528	136.460
Time per stock	10.065	9.622	8.319	12.476	16.718	4.549

5. Conclusion

This paper proposes a new deep LSTM network optimized by the ARO algorithm. The proposed model is named LSTM-ARO. The aim of the proposed model is to predict stock market prices. DJIA index stock price data is used to predict stock market prices. The dataset consists of 30 different stock prices between 2018.01.01 and 2022.12.31. The period of the data is five years. The data is converted to a new format that has 20 previous days for predicting the next day's price. After converting to the new format, the data is split 80% as train data and 20% as test data. A novel LSTM network is created with parameters. The parameters are connected to the ARO algorithm variables. When the variables change, the LSTM architecture changes. ARO finds the best architecture to get the best results. To evaluate the quality of the new proposed model, it is compared with LSTM1D, LSTM2D, LSTM3D, ANN, and LSTM-GA networks. The results clearly show that the LSTM-ARO model is the best model among the models. And it gives very successful predictions. The predictions can assist traders or investors.

For future research, the prediction system can be improved. Alternative metaheuristic algorithms can be tried. This system can be integrated into a dynamic trading system or an automated stock market system. It is aimed to use the successful model of this paper in the dynamic trading system. So, future values of the stocks can be predicted with low error. The trading system can give successful buy/sell/keep suggestions.

Declaration of Competing Interest

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

Data availability

The data is collected from Yahoo Finance (Yahoo Finance, 2023).

Acknowledgment

The author thanks TUBITAK (The Scientific and Technological Research Council of Türkiye) for using TRUBA (High Performance and Grid Computing Center).

References

Alkafaween, E., Hassanat, A. B., & Tarawneh, S. (2021). Improving Initial Population for Genetic Algorithm using the Multi Linear Regression Based Technique (MLRBT). *Communications-Scientific Letters of the University of Zilina, 23*(1), E1–E10.

Ashta, A., & Herrmann, H. (2021). Artificial intelligence and fintech: An overview of opportunities and risks for banking, investments, and microfinance. *Strategic Change, 30*(3), 211–222.

Askarzadeh, A., & Rezazadeh, A. (2013). Artificial neural network training using a new efficient optimization algorithm. *Applied Soft Computing, 13*(2), 1206–1213. https://doi.org/10.1016/j.asoc.2012.10.023

Bosworth, B., Hymans, S., & Modigliani, F. (1975). The stock market and the economy. *Brookings Papers on Economic Activity, 1975*(2), 257–300.

Chatfield, C. (2003). *The analysis of time series: An introduction*. Chapman and hall/CRC.

Chen, M.-Y., Liao, C.-H., & Hsieh, R.-P. (2019). Modeling public mood and emotion: Stock market trend prediction with anticipatory computing approach. *Computers in Human Behavior, 101*, 402–408. https://doi.org/10.1016/j.chb.2019.03.021

Garita, M. (2021). Using Stock Market Data in Python. In M. Garita (Ed.), *Applied Quantitative Finance: Using Python for Financial Analysis* (pp. 71–83). Springer International Publishing. https://doi.org/10.1007/978-3-030-29141-9_4.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Computation, 12*(10), 2451–2471. https://doi.org/10.1162/089976600300015015

Giudici, P., Agstner, P., & Capizzi, A. (2022). The Corporate Design of Investments in Startups: A European Experience. *European Business Organization Law Review*, 1–34.

Gornall, W., & Strebulaev, I. A. (2022). The Contracting and Valuation of Venture Capital-Backed Companies. *Forthcoming, Handbook of the Economics of Corporate. Finance, 1*.

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. (2021). Prediction of retail prices of roasted coffee by time series analysis. *International Anatolian Congress on Coffee and Cocoa*, 63.

Gülmez, B. (2022a). *MonkeypoxHybridNet: A hybrid deep convolutional neural network model for monkeypox disease detection* (Vol. 3,, 49–64.

Gülmez, B. (2022b). Zaman serisi analizi ile talep tahmini ve bir fabrikadaki üretim planlama, *Vol. 2. Mühendislik Alanında Uluslararası Araştırmalar* (pp. 57–74). Egitim Yayinevi.

Gülmez, B. (2022c). A novel deep neural network model based Xception and genetic algorithm for detection of COVID-19 from X-ray images. *Annals of Operations Research*. https://doi.org/10.1007/s10479-022-05151-y

Gülmez, B. (2022a). Yapay Tavşan Algoritmasıyla araç rotalama problemi optimizasyonu. *International Conference on Smart Logistics*, 86–87.

Gülmez, B. (2023a). Market zinciri ürün dağıtımı probleminin farklı genetik algoritma versiyonları ile çözümü ve karşılaştırması. *Osmaniye Korkut Ata Üniversitesi Fen Bilimleri Enstitüsü Dergisi, 6*(1), 180–196. https://doi.org/10.47495/okufbed.1117220

Gülmez, B. (2023). İşyeri güvenliği için derin öğrenme ile baret takılması tespiti. *International Conference on Engineering and Applied Natural Sciences*, 284.

Gülmez, B., & Korhan, E. (2021, November). *Covid-19 vaccine distribution time optimization with Genetic Algorithm*. The International Conference On Engineering, Natural And Applied Science (ICENAS 2021), Osmaniye, Turkey.

Gülmez, B. (2023a). A novel deep learning model with the Grey Wolf Optimization algorithm for cotton disease detection. *Journal of Universal Computer Science, 6*(29).

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

Gürbüz, S., & Şahbaz, A. (2022). Investigating the volatility spillover effect between derivative markets and spot markets via the wavelets: The case of Borsa İstanbul. *Borsa Istanbul Review, 22*(2), 321–331.

Hamilton, J. D. (2020). *Time Series Analysis*. Princeton University Press.

Hammoudeh, S., Mokni, K., Ben-Salha, O., & Ajmi, A. N. (2021). Distributional predictability between oil prices and renewable energy stocks: Is there a role for the COVID-19 pandemic? *Energy Economics, 103*(C). https://ideas.repec.org//a/eee/eneeco/v103y2021ics0140988321003947.html.

Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep learning. *Applied System Innovation, 4*(1), 9.

Kim, A., Yang, Y., Lessmann, S., Ma, T., Sung, M.-C., & Johnson, J. E. V. (2020). Can deep learning predict risky retail investors? A case study in financial risk behavior forecasting. *European Journal of Operational Research, 283*(1), 217–234. https://doi.org/10.1016/j.ejor.2019.11.007

Krogh, A. (2008). What are artificial neural networks? *Nature Biotechnology, 26*(2), 195–197.

Kumar, K., & Haider, M. T. U. (2021). Enhanced prediction of intra-day stock market using metaheuristic optimization on RNN–LSTM network. *New Generation Computing, 39*, 231–272.

Li, Y., Bu, H., Li, J., & Wu, J. (2020). The role of text-extracted investor sentiment in Chinese stock price prediction with the enhancement of deep learning. *International Journal of Forecasting, 36*(4), 1541–1562. https://doi.org/10.1016/j.ijforecast.2020.05.001

Li, Y., Zheng, W., & Zheng, Z. (2019). Deep robust reinforcement learning for practical algorithmic trading. *IEEE Access, 7*, 108014–108022.

Lim, H., & Rokhim, R. (2020). Factors affecting profitability of pharmaceutical company: An Indonesian evidence. *Journal of Economic Studies, 48*(5), 981–995. https://doi.org/10.1108/JES-01-2020-0021

Maguluri, L., & Ragupathy, R. (2020). An efficient stock market trend prediction using the real-time stock technical data and stock social media data. *Int. J. Intell. Eng. Syst, 13*(4), 316–332.

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), 2020*, 447–453.

Milana, C., & Ashta, A. (2021). Artificial intelligence techniques in finance and financial markets: A survey of the literature. *Strategic Change, 30*(3), 189–209.

Nguyen, T. N. L., & Nguyen, V. C. (2020). The determinants of profitability in listed enterprises: A study from Vietnamese stock exchange. *The Journal of Asian Finance, Economics and Business, 7*(1), 47–58.

Öztürk, M. M. (2022). Initializing hyper-parameter tuning with a metaheuristic-ensemble method: A case study using time-series weather data. *Evolutionary Intelligence*, 1–13.

Park, D.-Y., & Lee, K.-H. (2021). Practical algorithmic trading using state representation learning and imitative reinforcement learning. *IEEE Access, 9*, 152310–152321.

Rezaei, H., Faaljou, H., & Mansourfar, G. (2021). Stock price prediction using deep learning and frequency decomposition. *Expert Systems with Applications, 169*, Article 114332. https://doi.org/10.1016/j.eswa.2020.114332

Shah, D., Isah, H., & Zulkernine, F. (2019). Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies, 7*(2), 26.

Tao, R., Su, C.-W., Xiao, Y., Dai, K., & Khalid, F. (2021). Robo advisors, algorithmic trading and investment management: Wonders of fourth industrial revolution in financial markets. *Technological Forecasting and Social Change, 163*, Article 120421.

Teweles, R. J., & Bradley, E. S. (1998). *The stock market* (Vol. 64). John Wiley & Sons.

Théate, T., & Ernst, D. (2021). An application of deep reinforcement learning to algorithmic trading. *Expert Systems with Applications, 173*, Article 114632.

Wang, L., Cao, Q., Zhang, Z., Mirjalili, S., & Zhao, W. (2022). Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Engineering Applications of Artificial Intelligence, 114*, Article 105082. https://doi.org/10.1016/j.engappai.2022.105082

Xie, C., Rajan, D., & Chai, Q. (2021). An interpretable Neural Fuzzy Hammerstein-Wiener network for stock price prediction. *Information Sciences, 577*, 324–335. https://doi.org/10.1016/j.ins.2021.06.076

Yahoo Finance. (2023). Yahoo Finance. Quotes, Business and Finance News: Yahoo Finance - Stock Market Live. https://finance.yahoo.com/.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Computation, 31*(7), 1235–1270. https://doi.org/10.1162/neco_a_01199