

<sup>1</sup>Mohammed  
Kareem Ghadhbhan

<sup>2</sup>Shahab Jahangiri

<sup>3</sup>Ramin Bashir  
Khodaparasti

<sup>4</sup>Yousef  
Mohammadzadeh

## Enhancing Stock Price Forecasting with LSTM and GA: A Study on the Iraqi Market



**Abstract:** - This study investigates the application of the Long Short-Term Memory (LSTM) model, optimized through a Genetic Algorithm (GA), to predict stock prices of 20 selected companies listed on the Iraqi Stock Exchange. Using daily stock data spanning from 2020 to 2023—a period marked by significant economic and geopolitical turbulence—we assess the performance of the model in an emerging market characterized by low liquidity, high volatility, and sensitivity to political instability. The GA is employed to optimize key LSTM hyperparameters, such as optimizer type, loss function, activation function, dropout rate, and sequence size, in order to improve prediction accuracy. Results show that the LSTM model performs well for most stocks, with low error metrics and high  $R^2$  values for companies with less volatile stock prices, such as UQTN, HSAD, and MTCI. However, for more volatile stocks like ZAWK and BINV, prediction errors are significantly higher, indicating the need for more complex models or further refinements in parameter tuning. Overall, the LSTM-GA approach demonstrates strong potential for predictive modeling in emerging markets, offering valuable insights for investors and policymakers navigating the Iraqi stock market.

**Keywords:** LSTM, Genetic Algorithm, Stock Price Prediction, Iraqi Stock Exchange

### I. INTRODUCTION

In recent years, predicting stock prices has become one of the most challenging tasks in financial markets due to the dynamic and complex nature of the market. Stock price movements are influenced by various factors, including market trends, economic indicators, investor sentiment, and external shocks, making it difficult to develop models that can consistently forecast future prices. The rise of artificial intelligence (AI) has introduced new approaches to financial forecasting, particularly in the use of deep learning models like Long Short-Term Memory (LSTM) networks. LSTMs, a type of recurrent neural network (RNN), are well-suited for time series data due to their ability to capture long-term dependencies and trends, making them a popular choice for stock price prediction (Hewamalage et al., 2021; Fischer & Krauss, 2018).

LSTM models have demonstrated strong potential in capturing the temporal dependencies inherent in financial time series data, which includes stock prices. Compared to traditional statistical models like ARIMA or GARCH, LSTMs can learn non-linear patterns and complex relationships within the data (Zhou et al., 2022). This makes them advantageous in financial markets where sudden changes, volatility, and non-linearity are common. However, the performance of LSTM models in predicting stock prices largely depends on the careful selection of model parameters, such as the number of hidden layers, learning rate, and batch size (Goodfellow et al., 2016). Optimizing these hyperparameters can significantly improve model accuracy and generalizability.

One innovative approach to optimizing LSTM models is the use of a Genetic Algorithm (GA). GA is a biologically inspired optimization technique based on the principles of natural selection and evolution (Holland, 1992). It is particularly effective in solving complex optimization problems by iteratively evolving a population of solutions. When applied to LSTM hyperparameter tuning, GA can explore a wide search space to identify the best combination of parameters, thereby enhancing the model's performance (Mirjalili et al., 2020). Studies have shown that GA-based optimization can outperform traditional grid search or random search methods in finding optimal LSTM parameters for financial forecasting (Xiong et al., 2022; Aljarah et al., 2018).

In this study, we apply the LSTM model with GA-based optimization to predict the stock prices of 20 selected companies listed on the Iraqi Stock Exchange. The data consists of daily stock prices from the beginning of 2020 to the end of 2023, a period marked by significant economic and geopolitical events. The Iraqi stock market, like other emerging markets, exhibits unique characteristics such as lower liquidity, higher volatility, and sensitivity to political instability (Abdullah & Masih, 2020). As such, developing a robust predictive model that captures these dynamics is crucial for investors and policymakers.

### Review of Literature

The theoretical basis of using Long Short-Term Memory (LSTM) networks in stock price prediction lies in the ability of deep learning models to capture complex, non-linear relationships and long-term dependencies in time series data. Stock prices are inherently sequential, where current values are dependent on past prices and other market conditions. Traditional statistical models, such as autoregressive integrated moving average (ARIMA) or generalized autoregressive conditional heteroskedasticity (GARCH), often struggle with non-linearity and sudden shifts in financial markets (Zhou

et al., 2022). In contrast, LSTMs, a type of recurrent neural network (RNN), are specifically designed to handle these challenges by maintaining an internal memory that tracks information over time (Hochreiter & Schmidhuber, 1997).

### **LSTM Networks for Time Series Prediction**

LSTMs are uniquely suited for financial forecasting due to their architecture, which includes memory cells, input gates, output gates, and forget gates. This structure allows LSTMs to selectively retain or forget past information, addressing the vanishing gradient problem commonly associated with standard RNNs (Goodfellow et al., 2016). The vanishing gradient problem occurs when early information in a sequence gets lost as the model processes more time steps, reducing the model's ability to learn from long-term dependencies. LSTMs overcome this by retaining relevant past information while discarding irrelevant details, making them particularly powerful for stock price prediction where long-term dependencies may be critical (Fischer & Krauss, 2018).

In financial markets, prices often exhibit long-term patterns influenced by macroeconomic factors, investor sentiment, and market volatility. LSTMs can model these intricate dependencies over time and can outperform traditional models that assume linear relationships (Zhou et al., 2022). Numerous studies have validated the effectiveness of LSTMs in predicting stock prices, showing that deep learning-based methods offer superior performance, especially in volatile or non-linear market conditions (Hewamalage et al., 2021).

### **Genetic Algorithms for Parameter Optimization**

The performance of LSTM models is highly dependent on the tuning of hyperparameters, such as the number of hidden units, learning rate, batch size, and dropout rate (Goodfellow et al., 2016). Traditionally, these hyperparameters are chosen using manual tuning methods such as grid search or random search. However, these methods are computationally expensive and may not yield optimal results, especially for complex models like LSTM, which have many parameters to adjust (Bergstra & Bengio, 2012).

Genetic Algorithms (GAs) provide a more efficient solution for hyperparameter optimization. GAs are based on the principles of natural selection and evolution, where a population of potential solutions evolves over time to find the optimal set of parameters (Holland, 1992). In the context of LSTM optimization, GAs can be used to explore the hyperparameter space more efficiently, adjusting parameters like the learning rate, the number of epochs, and the network's depth by selecting the most "fit" combinations of parameters in each generation (Mirjalili et al., 2020).

When combined, LSTMs and GAs offer a powerful approach for stock price prediction. GAs can optimize the hyperparameters of the LSTM model to enhance its ability to capture market trends and volatility (Xiong et al., 2022). Research has shown that GA-optimized LSTM models outperform those tuned using traditional methods, resulting in higher prediction accuracy and better generalizability to unseen data (Aljarah et al., 2018). This optimization is particularly important in financial markets where high volatility and sudden shifts in prices require a model that is both flexible and robust (Zhou et al., 2022).

### **Application to Stock Price Prediction**

The application of LSTM models optimized by GAs to stock price prediction, particularly in the context of the Iraqi Stock Exchange, provides a unique case study. Emerging markets like Iraq's often experience high levels of volatility, lower liquidity, and greater sensitivity to geopolitical events, which makes accurate stock price prediction more challenging (Abdullah & Masih, 2020). The dynamic nature of such markets requires models that can not only capture immediate price fluctuations but also longer-term trends influenced by external factors. LSTMs, with their ability to maintain and forget information over different time intervals, are well-suited for this task, while GA optimization ensures that the model's parameters are fine-tuned to maximize prediction accuracy.

Given the daily stock price data from 2020 to 2023, this study aims to utilize LSTM's deep learning capabilities to capture the underlying patterns in stock price movements while leveraging GAs to optimize the model's performance. The combination of these two advanced AI techniques offers a novel approach to financial forecasting, particularly in an emerging market context where traditional models may struggle to deliver reliable predictions.

### **Empirical Studies**

An empirical literature review on the application of Long Short-Term Memory (LSTM) models, optimized with Genetic Algorithms (GA), for stock price prediction highlights the significant advancements in financial forecasting using artificial intelligence (AI) techniques. This review synthesizes studies that empirically demonstrate the effectiveness of LSTMs and GAs in predicting stock prices, focusing on their application across different markets, datasets, and modeling frameworks.

### **LSTM Networks in Stock Price Prediction**

LSTM models have become a prominent tool for stock price prediction due to their ability to model long-term dependencies and non-linear patterns in time series data. Empirical research consistently highlights LSTM's superior performance over traditional models in capturing the complexity of stock market data.

For instance, Fischer and Krauss (2018) conducted one of the earliest studies applying LSTMs to stock market predictions. They used a large dataset of S&P 500 stocks to show that LSTM networks outperformed logistic regression and random forests in predicting stock returns. Their results demonstrated that LSTM networks could model temporal dependencies effectively, making them ideal for financial time series forecasting.

A more recent study by Hewamalage et al. (2021) conducted a comparative analysis of LSTM models and other recurrent neural networks (RNNs) for time series forecasting. Using multiple financial datasets, including stock prices, they found that LSTM models consistently outperformed traditional RNNs, convolutional neural networks (CNNs), and simple moving averages. The authors attributed the success of LSTMs to their architecture, which mitigates the vanishing gradient problem and allows for the capture of long-term dependencies in volatile financial markets.

Empirical studies in emerging markets also provide evidence of the effectiveness of LSTMs. Zhou et al. (2022) applied LSTM models to predict stock prices in China's stock market and found that LSTMs provided more accurate predictions compared to ARIMA and support vector machines (SVMs). This study highlights the broad applicability of LSTM models across various markets, both developed and emerging, where market volatility and non-linearity pose significant challenges for traditional models.

### **Genetic Algorithm (GA) Optimization for LSTM Models**

The application of Genetic Algorithms (GAs) for optimizing LSTM models has emerged as a promising approach for enhancing predictive performance. GAs provide an efficient way to optimize the hyperparameters of LSTM networks, such as the number of hidden layers, learning rate, and batch size, which are critical to the model's accuracy.

Xiong et al. (2022) conducted an empirical study using GAs to optimize LSTM models for stock price prediction. Their study focused on the U.S. stock market and compared the GA-optimized LSTM model with standard LSTM and random search methods. The GA-optimized model demonstrated superior performance, particularly in improving prediction accuracy and minimizing overfitting. The authors argued that the stochastic nature of financial markets requires models with hyperparameters that adapt to different market conditions, which GAs can achieve more efficiently than traditional optimization methods.

Similarly, Aljarah et al. (2018) employed a hybrid approach that combined LSTM with a Whale Optimization Algorithm (WOA), a meta-heuristic inspired by genetic evolution, to forecast stock prices. Their empirical results, based on historical stock data from the Amman Stock Exchange, showed that optimization algorithms like GAs and WOA significantly enhance the predictive power of LSTM models by finding optimal hyperparameters. This is especially important in financial markets where small improvements in prediction accuracy can lead to substantial financial gains.

Mirjalili et al. (2020) provided a comprehensive review of evolutionary algorithms, including GAs, in optimizing deep learning models. Their findings supported the notion that GAs are effective for hyperparameter tuning in LSTM models, particularly in high-dimensional and complex data environments like financial time series. They emphasized that GA's ability to search a wide parameter space with fewer computational resources compared to grid or random search makes it highly suitable for stock market forecasting.

### **LSTM and GA in Emerging Markets**

Emerging markets, characterized by higher volatility and lower liquidity, pose unique challenges for stock price prediction. Several empirical studies have applied LSTM and GA models to emerging markets, demonstrating the robustness of these AI techniques in volatile environments.

For example, Abdullah and Masih (2020) studied volatility spillovers and stock price prediction in the Iraqi Stock Exchange using deep learning models. Their findings indicated that LSTMs, combined with optimization techniques like GAs, were highly effective in predicting stock price movements in the Iraqi market, which is heavily influenced by geopolitical risks and external shocks. They reported that GA-optimized LSTM models could capture both short-term fluctuations and long-term trends better than traditional econometric models such as GARCH.

A study by Alharbi et al. (2021) focused on stock price prediction in the Saudi Arabian stock market, another emerging market with high volatility. They applied LSTM networks optimized using GAs and found that the models could accurately forecast stock price movements during periods of both low and high volatility. The study showed that in emerging markets, where market conditions can shift dramatically, GA-optimized LSTM models provided better adaptability and resilience to sudden market changes.

These studies suggest that GA-optimized LSTM models are particularly well-suited for emerging markets due to their ability to manage complex and volatile financial data. By fine-tuning hyperparameters using GAs, researchers have consistently achieved improved forecasting accuracy, providing valuable insights for investors and policymakers in these markets.

### Comparative Studies: LSTM and Traditional Models

Several studies have empirically compared the performance of LSTM models, both with and without GA optimization, to traditional statistical models in stock price prediction. These comparisons consistently favor LSTM-based approaches.

For example, Sezer et al. (2020) conducted a large-scale empirical analysis comparing deep learning models, including LSTM, with traditional models such as ARIMA and SVM for stock price prediction across multiple global markets. Their results indicated that LSTM models, especially when optimized using GAs, outperformed traditional models in terms of prediction accuracy and ability to capture non-linear dependencies in the data. The study concluded that the LSTM-GA hybrid approach offers the most robust solution for financial time series forecasting.

A similar study by Zhang et al. (2021) focused on the performance of LSTM models compared to econometric models like GARCH for predicting stock market volatility in the Indian stock market. The findings revealed that the LSTM model, particularly when hyperparameters were optimized using GAs, provided more accurate predictions of stock volatility. The study highlighted that LSTM's ability to model non-linear dependencies gave it an edge over traditional models, which often rely on assumptions of linearity.

## II. METHOD

The methodology for using Long Short-Term Memory (LSTM) networks, optimized by Genetic Algorithms (GA) for stock price prediction, involves a sequence of technical steps designed to prepare the data, develop the model, optimize hyperparameters, and evaluate performance. Below is a detailed breakdown of each phase:

### 1. Data Collection and Preprocessing

The study uses daily stock price data from 20 selected companies on the Iraqi Stock Exchange, spanning from the beginning of 2020 to the end of 2023. The dataset typically includes variables such as the opening price, closing price, high price, low price, and trading volume.

#### a. Data Collection:

- **Data Sources:** The stock data can be retrieved from financial databases, the stock exchange itself, or third-party APIs.
- **Data Frequency:** Daily stock prices, meaning one observation per trading day per stock.
- **Data Variables:** Time, stock price (open, high, low, close), volume, and potentially other technical indicators like moving averages or volatility indices.

#### b. Data Preprocessing:

- **Handling Missing Data:** Any missing data is either imputed using statistical methods like forward or backward filling or removed if the missingness is significant.
- **Normalization:** Stock price data is normalized using techniques like Min-Max scaling or Z-score normalization to ensure all input features are on a similar scale, which is crucial for the convergence of the LSTM model. Min-Max scaling transforms data into a range [0,1] using:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

- **Train-Test Split:** The data is divided into training and testing sets, typically using a 70/30 or 80/20 split. Given the nature of financial data, this is done chronologically to avoid data leakage (i.e., future information entering the training set).

- **Windowing:** Since LSTMs require sequential data, a sliding window technique is used to create sequences of input-output pairs. For example, a window of 10 days might predict the stock price on the 11th day:
  - Input: Stock prices from day 1 to day 10.
  - Output: Stock price on day 11.

## 2. LSTM Model Architecture

LSTM is a specialized recurrent neural network (RNN) designed to model sequential data. In this case, the model learns from past stock prices to predict future values.

### a. LSTM Cell:

Each LSTM cell consists of four primary components:

- **Forget Gate:** Decides what information from the previous time step is discarded.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

**Input Gate:** Determines what new information is stored in the cell state.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

**Cell State Update:** Updates the cell state based on input.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

**Output Gate:** Determines what part of the cell state should be output.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot \tanh(C_t)$$

Where  $h_t$  is the hidden state,  $C_t$  is the cell state,  $x_t$  is the input at time step  $t$ , and  $\sigma$  represents the sigmoid activation function.

### b. Model Layers:

- **Input Layer:** Takes in the normalized time-series data with features representing stock prices and other indicators.
- **LSTM Layers:** One or more LSTM layers to capture the sequential dependencies in stock price data.
  - **Number of Units:** Defines the number of LSTM cells in each layer, a hyperparameter optimized later using GA.
- **Dense Layer (Fully Connected Layer):** A dense layer maps the final hidden states from the LSTM to a single output, representing the predicted stock price.
  - **Activation:** Typically, linear for regression tasks.

**c. Loss Function:**

The **Mean Squared Error (MSE)** is the most common loss function used for regression tasks like stock price prediction:

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

Where  $y_i$  is the actual stock price, and  $\hat{y}_i$  is the predicted stock price.

**d. Optimizer:**

An optimization algorithm like Adam (Adaptive Moment Estimation) is commonly used to minimize the loss function by adjusting model weights.

**3. Genetic Algorithm for Hyperparameter Optimization**

Genetic Algorithms (GAs) are applied to optimize the hyperparameters of the LSTM model. Key hyperparameters include the number of LSTM units, learning rate, dropout rate, and batch size. These parameters are difficult to tune manually, and GAs provide an efficient mechanism to search for the optimal combination.

**a. Initialization:**

The GA begins by randomly initializing a population of potential solutions, where each solution is a different set of hyperparameters (e.g., the number of units, learning rate). Each set of hyperparameters is encoded as a “chromosome.”

**b. Fitness Function:**

Each chromosome is evaluated based on the performance of the corresponding LSTM model on the training set. The fitness function could be the inverse of the model's loss (e.g.,  $1/MSE$ ) or another metric like Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

This metric measures the accuracy of the model predictions during training.

**c. Selection:**

Chromosomes are selected for reproduction based on their fitness scores. Selection techniques like roulette wheel selection or tournament selection are used to ensure that better-performing models have a higher chance of being selected.

**d. Crossover and Mutation:**

- **Crossover:** Two selected chromosomes are combined to produce offspring by swapping parts of their hyperparameter values.
- **Mutation:** Random alterations are introduced to some hyperparameter values in the offspring to ensure diversity in the search space.

**e. Iteration and Convergence:**

The GA process iterates over several generations, with each generation producing new offspring and retaining the best solutions from the previous generation. The process continues until a predefined convergence criterion is met, such as a maximum number of generations or a fitness threshold.

#### 4. Model Evaluation and Validation

Once the LSTM model is optimized using GA, it is evaluated on the test set (unseen data). The performance is measured using several key metrics:

##### a. Performance Metrics:

- **Mean Squared Error (MSE):** Measures the average squared difference between actual and predicted stock prices.
- **Root Mean Squared Error (RMSE):** Provides an interpretable measure of error in the same units as the stock prices.
- **Mean Absolute Percentage Error (MAPE):** Evaluates the percentage error between actual and predicted prices:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100$$

**R-squared (R<sup>2</sup>):** Measures the proportion of the variance in the dependent variable (stock price) that is predictable from the independent variables (features):

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

### III. DATA AND RESEARCH FINDINGS

This paper utilizes the stock price data of 20 selected companies listed on the Iraqi Securities Exchange, covering the period from the beginning of 2020 to the end of 2023. The names of these companies are provided in Table 1.

**Table 1: Titles of selected companies and their symbols**

stock	symbol	stock	symbol	stock	symbol	stock	symbol
Al-Mansour Bank	<b>BMNS</b>	South Gas Company	<b>SGAS</b>	Iraqi International Law Group	<b>IILG</b>	Baghdad Investment	<b>BINV</b>
Bank of Baghdad	<b>BBOB</b>	Asiacell Communications	<b>TASC</b>	Erbil Cement Company	<b>ECRC</b>	Al-Sadeer Hotel	<b>HSAD</b>
Iraqi Airways	<b>IAIR</b>	National Bank of Iraq	<b>BNOI</b>	Al-Naft Petroleum Company	<b>INPC</b>	Mobile Teleco.. Company	<b>MTCI</b>
Al-Quytoon General Trading	<b>UQTN</b>	Al-Ahlyia for Agricultural Production	<b>AAPC</b>	Gulf Commercial Bank	<b>BGUC</b>	Al-Tamimi for Real Estate Investment	<b>TRIM</b>
Baghdad Soft Drinks	<b>IBSD</b>	Middle East Investment Bank	<b>MIBI</b>	Mosul Bank	<b>BMFI</b>	Al-Zawraa for Poultry Products	<b>ZAWK</b>

In this section, the results related to the LSTM algorithm with parameter optimization by GA are presented separately for stock exchange symbols. For this purpose, the optimal parameters and evaluation results are presented separately for the 20 investigated symbols. In this research, Python programming language and specially two popular libraries, keras and tensorflow, were used to train the LSTM model.

The data used in this research was divided into 70% for training and 30% for testing. Considering that LSTM model training has a set of parameters. For this purpose, the obtainable values of these parameters are presented in Table 2. Also, the values of adjustable parameters in GA are presented in Table 3.

**Table 2: Adjustable parameter values in LSTM algorithm**

Adjustable parameter	values
Optimizer	['RMSprop', 'Adam', 'Adamax', 'Nadam']
Lose	[mse, 'mae', 'mape', 'msle']
Activation	['relu', 'linear']
Dropout	[0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5]
sequence size	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
Neuron	[8, 16, 32, 64, 128, 256, 512]
Batch	[8, 16, 32, 64, 128]
Epoch	[50, 100, 150, 200]
kernel initializer	['random_normal', 'uniform', 'random_uniform', 'he_normal']

**Table 3: Values of adjustable parameters in GA**

Adjustable parameter	values
Initial population size	60
Dimensions of population members	9
Number of repetitions	100
Combination factor	0.95
Mutation coefficient	0.05

In this study, a Genetic Algorithm (GA) was employed during the LSTM model training process to optimize specified parameter values (Table 2) and identify the best configuration. As a result, the optimal parameters were selected from the given meta-parameters. For each of the 20 analyzed stock symbols, the optimal adjustable LSTM parameters (Table 4) and the corresponding accuracy evaluation metrics (Table 5) are presented based on the research data set.

**Table 4: Optimum values of adjustable parameters in LSTM algorithm with GA for stock symbols**

symbol	Optimizer	lose	activation	dropout	sequence size	neuron	batch	epoch	kernel initializer
<b>BMNS</b>	Nadam	mape	linear	0	12	64	64	200	uniform
<b>BBOB</b>	Adamax	mae	linear	0	2	64	32	50	he_normal
<b>IAIR</b>	Nadam	mape	linear	0.4	2	512	16	200	uniform
<b>UQTN</b>	Nadam	mse	linear	0	11	512	8	200	uniform
<b>IBSD</b>	Nadam	mape	linear	0	19	64	16	200	he_normal
<b>SGAS</b>	RMSprop	mape	linear	0.1	4	256	16	100	uniform
<b>TASC</b>	Adamax	mae	relu	0.45	1	256	8	100	random_uniform
<b>BNOI</b>	RMSprop	msle	relu	0.45	4	512	32	200	he_normal
<b>AAPC</b>	Adam	mae	linear	0.45	4	128	32	200	he_normal
<b>MIBI</b>	RMSprop	mae	linear	0.2	2	64	16	200	uniform
<b>IILG</b>	Nadam	mape	linear	0	1	512	32	100	he_normal
<b>ECRC</b>	Adam	mape	linear	0.5	5	512	16	200	he_normal
<b>INPC</b>	Adam	msle	linear	0.25	2	128	8	50	he_normal
<b>BGUC</b>	Adam	mape	relu	0.45	7	256	16	150	he_normal
<b>BMFI</b>	Adamax	mae	linear	0.3	3	256	64	200	he_normal
<b>BINV</b>	Adamax	mae	relu	0.1	1	512	128	200	uniform
<b>HSAD</b>	Adam	mape	linear	0.4	5	512	64	50	he_normal
<b>MTCI</b>	Nadam	mape	linear	0.3	18	512	16	50	uniform
<b>TRIM</b>	Nadam	mape	linear	0.1	14	128	8	200	random_uniform
<b>ZAWK</b>	Adam	mape	linear	0.45	2	64	8	200	he_normal

This table presents the optimum values of adjustable parameters for the LSTM model optimized by Genetic Algorithms (GA) across different stock symbols. These parameters define how the LSTM models are configured to achieve the best possible performance for each stock symbol on the Iraqi Stock Exchange. Below is an interpretation of the key parameters and their significance:

### 1. Symbol

Each row in the table represents a different stock symbol (e.g., BMNS, BBOB, IAIR). These symbols correspond to 20 selected companies on the Iraqi Stock Exchange. For each symbol, a set of hyperparameters has been optimized using GA to enhance the performance of the LSTM model for predicting stock prices.

### 2. Optimizer

- Nadam, Adamax, RMSprop, Adam: These are optimization algorithms that adjust the learning rates during model training.
  - Nadam: A variant of Adam that incorporates Nesterov momentum.
  - Adamax: A variant of Adam based on the infinity norm.
  - RMSprop: Root Mean Square Propagation, often used in recurrent neural networks like LSTMs.
  - Adam: A popular optimization algorithm that adapts learning rates for each parameter.

The choice of optimizer influences how quickly and accurately the model converges. For example, BMNS and IAIR use Nadam, whereas BBOB uses Adamax, suggesting that different optimizers work better for different stock symbols.

### 3. Loss Function

- mape (Mean Absolute Percentage Error)
- mae (Mean Absolute Error)
- mse (Mean Squared Error)
- msle (Mean Squared Logarithmic Error)

These are the error metrics the model minimizes during training. For example:

- MAPE is often used for time series data as it gives a percentage error.
- MAE measures the average magnitude of errors.
- MSE emphasizes larger errors.
- MSLE penalizes underpredictions more than overpredictions.

For instance, BMNS and IAIR models aim to minimize MAPE, which focuses on percentage accuracy, while BNOI uses MSLE to account for logarithmic differences.

### 4. Activation Function

- linear and relu (Rectified Linear Unit)

The activation function defines the output of neurons in the LSTM network.

- Linear activation is used when predicting continuous values, such as stock prices.
- ReLU is more common in hidden layers and helps to handle non-linearities.

For example, most stocks use linear activation, but stocks like TASC and BGUC use ReLU, likely because these models benefit from non-linear feature interactions in their data.

## 5. Dropout Rate

The dropout rate indicates the proportion of neurons that are randomly "dropped" or ignored during training to prevent overfitting. A higher dropout rate adds regularization to avoid overfitting.

- Dropout rates vary from 0 (no dropout) in stocks like BMNS and IBSD to 0.5 (50%) in ECRC and INPC. This suggests that some stocks require more regularization to avoid overfitting, possibly due to more volatile or noisy data.

## 6. Sequence Size

This parameter defines the number of time steps (previous days) the model uses as input for making predictions. Larger sequence sizes help the model capture longer-term dependencies but can also make training more complex.

- For example, BMNS uses a sequence size of 12, meaning it uses data from the last 12 days to predict the stock price.
- On the other hand, TASC uses a sequence size of just 1, indicating the model relies on very recent data for predictions.

## 7. Neuron Count

The number of neurons in the LSTM layer defines the model's capacity to learn from the data. More neurons generally allow the model to capture more complex patterns but may increase the risk of overfitting and computational cost.

- For example, BMNS and BBOB use 64 neurons, while IAIR and UQTN use 512 neurons, suggesting that stocks like IAIR require more model complexity to capture underlying patterns.

## 8. Batch Size

Batch size refers to the number of training samples the model processes before updating its weights. Smaller batch sizes can lead to noisier gradient updates, while larger ones provide more stable updates.

- Stocks like BMNS and BMFI use a batch size of 64, indicating larger batches, while stocks like UQTN use smaller batch sizes (e.g., 8), possibly due to data variability or computational considerations.

## 9. Epoch Count

Epochs define the number of times the model goes through the entire dataset during training. More epochs usually improve model performance but can lead to overfitting if too high.

- Most stock models are trained for 200 epochs, as seen for BMNS, IAIR, IBSD, etc., while some, like TASC and HSAD, are trained for fewer epochs (50–100), potentially to avoid overfitting or excessive computational cost.

## 10. Kernel Initializer

- uniform, he\_normal, random\_uniform: These methods initialize the weights of the LSTM model.
  - Uniform initializes weights within a uniform distribution.
  - He\_normal is used specifically for layers with ReLU activation to deal with vanishing gradients.
  - Random\_uniform randomly initializes weights uniformly.

For example, BMNS uses a uniform initializer, while BGUC uses he\_normal to support its ReLU activation function.

In Summary, the table illustrates that different stock symbols on the Iraqi Stock Exchange require specific LSTM configurations to achieve optimal prediction accuracy. Each model has been fine-tuned using Genetic Algorithms to adjust key hyperparameters such as optimizer, loss function, neuron count, dropout rate, and others, demonstrating the heterogeneity of stock behavior. The variations in activation functions, sequence sizes, and optimizers reflect the different market dynamics for each stock, with more complex models (e.g., larger neurons, smaller batch sizes) generally being used for stocks that likely exhibit more non-linear or volatile patterns.

Table 5 shows the accuracy evaluation indices including: MSE, RMSE, MAE, MAPE and R2 for the research dataset. As can be seen, they have obtained different accuracies in each stock symbol category.

**Table 5: The results of LSTM algorithm evaluation indices with GA for stock symbols**

symbol	MSE	RMSE	MAE	MAPE	R <sup>2</sup>
<b>BMNS</b>	7842.5373	88.5581	72.5359	0.0222	0.942290
<b>BBOB</b>	2815.0193	53.0568	42.6733	0.0217	0.933306
<b>IAIR</b>	2599.1571	50.9819	42.4303	0.0218	0.938514
<b>UQTN</b>	1318.4575	36.3106	28.2781	0.0144	0.981526
<b>IBSD</b>	1249.4849	35.3481	26.4250	0.0159	0.927223
<b>SGAS</b>	29803.0717	172.6357	111.0321	0.0141	0.946916
<b>TASC</b>	70103.3197	264.7703	207.8238	0.0210	0.791156
<b>BNOI</b>	18418.4189	135.7145	107.2015	0.0238	0.948928
<b>AAPC</b>	42102.2970	205.1884	153.2919	0.0144	0.887677
<b>MIBI</b>	378118.4769	614.9134	204.2443	0.0225	0.964705
<b>IILG</b>	672689.5841	820.1766	233.8386	0.0250	0.977590
<b>ECRC</b>	353883.9237	594.8814	470.1380	0.0203	0.973326
<b>INPC</b>	3615.6865	60.1306	50.1428	0.0255	0.855493
<b>BGUC</b>	2044.8540	45.2201	37.5392	0.0224	0.880773
<b>BMFI</b>	2760.7963	52.5433	43.2879	0.0238	0.836363
<b>BINV</b>	52343.3296	228.7866	201.7811	0.0287	0.874493
<b>HSAD</b>	240834.2687	490.7487	221.8824	0.0173	0.988143
<b>MTCI</b>	6330.4245	79.5640	58.0160	0.0181	0.990376
<b>TRIM</b>	12690.0765	112.6502	84.0002	0.0175	0.981440
<b>ZAWK</b>	3044094.7543	1744.7334	709.7510	0.0260	0.976497

This table presents the evaluation metrics for the performance of the LSTM model optimized by Genetic Algorithms (GA) for each stock symbol. These metrics include MSE (Mean Squared Error), RMSE (Root Mean Squared Error), MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), and R<sup>2</sup> (Coefficient of Determination). Let's break down the interpretation of each of these metrics and how they reflect the performance of the model across different stock symbols.

### 1. Mean Squared Error (MSE)

MSE measures the average squared difference between the predicted stock prices and the actual stock prices. Lower MSE values indicate better model performance.

- For example, UQTN has the lowest MSE of 1318.4575, indicating that the LSTM model is highly accurate in predicting the stock prices for this stock.
- On the other hand, ZAWK has an MSE of 3,044,094.7543, suggesting that the predictions for this stock are less accurate compared to others.

### 2. Root Mean Squared Error (RMSE)

RMSE is the square root of the MSE, which makes it easier to interpret since it is in the same units as the stock prices. Lower RMSE values indicate better prediction accuracy.

- UQTN also has the lowest RMSE of 36.3106, reinforcing its strong performance.
- ZAWK has the highest RMSE of 1,744.7334, reflecting poor prediction accuracy.

### 3. Mean Absolute Error (MAE)

MAE measures the average magnitude of the errors in the predictions, without considering their direction (i.e., whether the error is positive or negative). It is more interpretable than MSE and RMSE as it directly represents the average error in the same units as the stock prices.

- IBSD has the lowest MAE of 26.4250, meaning the average prediction error for this stock is approximately 26.43 units.
- ZAWK has the highest MAE of 709.7510, suggesting large discrepancies between the predicted and actual stock prices.

#### 4. Mean Absolute Percentage Error (MAPE)

MAPE expresses the error as a percentage of the actual stock prices, providing a relative measure of the model's performance. Lower MAPE values are better, especially when dealing with stocks at different price levels.

- SGAS and AAPC have the lowest MAPE values (0.0141 and 0.0144, respectively), indicating that the percentage error in predictions for these stocks is very low, making the models quite accurate.
- BINV and ZAWK have the highest MAPE values (0.0287 and 0.0260), showing higher relative prediction errors.

#### 5. R<sup>2</sup> (Coefficient of Determination)

R<sup>2</sup> indicates how well the model's predictions fit the actual stock prices. It ranges from 0 to 1, with values closer to 1 indicating better fit and higher explanatory power of the model.

- HSAD and MTCI have the highest R<sup>2</sup> values (0.988143 and 0.990376, respectively), indicating that the model explains a significant portion of the variance in the stock prices for these companies.
- TASC has the lowest R<sup>2</sup> of 0.791156, meaning the model is less effective in explaining the variability in the stock prices for this company.

## IV. CONCLUSION

The analysis of the results highlights the effectiveness and variability of using an LSTM model optimized with Genetic Algorithms (GA) for predicting stock prices on the Iraqi Stock Exchange. By adjusting the key hyperparameters of the LSTM model, such as the optimizer, loss function, activation function, dropout rate, sequence size, and the number of neurons, batch size, and epochs, the performance of the model has been fine-tuned for each stock symbol, leading to different outcomes across various stocks.

- **Model Performance Across Stock Symbols:** The results demonstrate that the LSTM model, when optimized with GA, can produce highly accurate predictions for some stocks, but there is significant variability in performance. Stocks like UQTN, HSAD, and MTCI consistently show excellent results across key evaluation metrics such as MSE, RMSE, MAE, MAPE, and R<sup>2</sup>. These stocks have low prediction errors and high explanatory power, with R<sup>2</sup> values nearing 1 (e.g., 0.990 for MTCI and 0.988 for HSAD). This suggests that for these stocks, the LSTM model has successfully captured the underlying market dynamics and patterns, leading to robust predictive capabilities.
- **Low and Moderate Prediction Errors:** Stocks such as BMNS, BBOB, and IAIR also exhibit good model performance, with relatively low MSE, RMSE, and MAE values, and R<sup>2</sup> values above 0.93. These results indicate that the model is able to effectively predict stock prices for these companies, though the error margins are slightly higher than the top-performing stocks. For example, BMNS has an MSE of 7,842.54, while BBOB has a lower MSE of 2,815.02. Both stocks still maintain high R<sup>2</sup> values of 0.942 and 0.933, respectively, showing a strong correlation between the model's predictions and the actual stock prices.
- **High-Error Predictions for Volatile Stocks:** In contrast, stocks such as ZAWK, BINV, and IILG exhibit poor predictive performance with much higher MSE, RMSE, and MAE values. ZAWK, in particular, stands out with the highest MSE (3,044,094.75) and RMSE (1,744.73), alongside a relatively lower but still strong R<sup>2</sup> value of 0.976. Despite the high R<sup>2</sup>, the large prediction errors suggest that these stocks exhibit significant volatility or complex patterns that the LSTM model, even when optimized, struggles to accurately predict. High MAPE values for stocks like BINV (0.0287) further support this conclusion, indicating that the percentage error in predictions is notably higher for these stocks.

- Impact of Optimized Parameters: The variability in performance across stocks can be traced back to the optimized LSTM parameters. Stocks like UQTN and IBSD, which exhibit the best performance, are configured with lower batch sizes, larger neurons, and uniform kernel initializers, which seem to suit their data patterns. In contrast, stocks with poorer performance, such as TASC and ZAWK, use configurations that, while optimized through GA, may not be sufficient to handle the volatility or noise in their price data. The dropout rates and sequence sizes also vary significantly, with higher dropout values used in some of the poorer-performing stocks, possibly indicating over-regularization or inadequate sequence lengths for capturing long-term dependencies.
- 5. The Role of the Genetic Algorithm in Optimization: The use of GA in optimizing LSTM parameters clearly enhances the model's adaptability to different stock symbols. By allowing the GA to explore various configurations, the LSTM model can find near-optimal solutions for specific stocks, as evidenced by the high-performing stocks with fine-tuned parameters. However, for stocks with more complex price movements, the current model may still require further refinements, perhaps through more sophisticated genetic algorithm settings, deeper networks, or hybrid models that combine LSTM with other techniques like attention mechanisms.

## V. REFERENCES:

- [1] Abdullah, A. M., & Masih, M. (2020). Volatility transmission and stock market dynamics in Iraq. *Journal of Financial Markets*, 23(1), 34-47.
- [2] Alharbi, M., Al-Sheikh, A., & Abdulaziz, M. (2021). Forecasting Saudi stock market trends using deep learning techniques. *International Journal of Economics and Financial Research*, 4(2), 234-245.
- [3] Aljarah, I., Faris, H., & Mirjalili, S. (2018). Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Computing*, 22(1), 1-15.
- [4] Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13, 281-305.
- [5] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2), 654-669.
- [6] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- [7] Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1), 388-427.
- [8] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
- [9] Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems*. MIT Press.
- [10] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2020). Genetic algorithm: A review of the literature and applications. *Evolutionary Computation*, 23(3), 56-69.
- [11] Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005-2019. *Applied Soft Computing*, 90, 106181.
- [12] Xiong, R., Nichols, E., & Chang, S. (2022). Stock price forecasting using deep learning and genetic algorithms: A comparative analysis. *Applied Soft Computing*, 112, 107846.
- [13] Zhang, W., Wei, X., & Zhang, Z. (2021). Predicting stock market volatility using LSTM networks with GA optimization. *Journal of Financial Markets*, 36, 101234.
- [14] Zhou, Z., Yin, Y., & Lin, Y. (2022). Stock market prediction using deep learning and hybrid models. *Expert Systems with Applications*, 180, 115040.