ISSN: 2093-5374 / E-ISSN: 2093-5382

Stock Index Prediction Using Temporal Convolutional Network and Long Short-Term Memory Network Optimized by Genetic Algorithm

Zeren Shi[1*], Othman Ibrahim[2], and Hanini Ilyana Che Hashim[3]

[1*]Faculty of Management, University Technology Malaysia, Skudai 81310, Johor, Malaysia.
shizeren@graduate.utm.my, https://orcid.org/0009-0003-1765-3024

[2]Faculty of Management, University Technology Malaysia, Skudai 81310, Johor, Malaysia.
othmanibrahim@utm.my, https://orcid.org/0000-0002-8384-4078

[3]Faculty of Management, University Technology Malaysia, Skudai 81310, Johor, Malaysia.
haniniilyana@utm.my, https://orcid.org/0000-0002-7732-8114

Received: November 13, 2024; Revised: December 23, 2024; Accepted: February 13, 2025; Published: March 31, 2025

Abstract

This research proposes the use of GA (Genetic Algorithm), LSTM (Long Short-Term Memory) and TCN (Temporal Convolutional Network) to predict movements in the stock index with the aid of a hybrid model, and TCN efficiently extracts models that have long-term dependencies as well as local features from time-series data, whereas LSTM captures nonlinear relationships and long- term trends. GA optimizes the hyperparameters of the model to enhance the accuracy of prediction. The hybrid GA-TCN-LSTM model is applied to several international stock markets and shows superior forecasting performance and profitability compared to traditional neural network models. Our study aids in enhancing stock market forecasting with the employment of a robust model which can adapt to a broad variety of market conditions, improving investment decisions for financial analysts and investors.

Keywords: TCN, LSTM, Genetic Algorithm, Stock Prediction.

1 Introduction

The stock market has a significant importance in the global economy. Characterized by its naturally noisy, non-parametric, non-linear, and deterministic chaotic systems (Ahangar et al., 2010). External factors like investor psychology, political events and the performance of various stock markets can affect the price of stocks (Atilgan et al., 2013). Accurately forecasting the prices of stock is significant so that financial analysts as well as investors can make proper decisions and increase returns. Traditional statistical methods often fall short of capturing the intricate patterns and nonlinear relationships inherent in stock market data (Jalaja et al., 2024). Methods like GBM (Gradient Boosting Machines), SVM (Support Vector Machines), RF (Random Forests) etc. have been applied with varying degrees of success (Hu et al., 2015; Patel et al., 2015; Sezer et al., 2020; Mabu et al., 2015; Dash & Dash, 2016). However, these methods still encounter drawbacks while modelling long-term dependencies as well as while dealing with the temporal aspect of financial Time Series Data (TSD) (Prabu et al., 2018).

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA), volume: 16, number: 1 (March), pp. 508-527. DOI: *10.58346/JOWUA.2025.I1.030*

*Corresponding author: Faculty of Management, University Technology Malaysia, Skudai 81310, Johor, Malaysia.

The rapid advancements in deep learning (DL) have spurred significant interest in developing novel algorithms for time-series forecasting, which have shown promise in improving the accuracy of making predictions in stock markets (Basnet et al., 2019). LSTMs can deal with nonlinearities and long-term dependent data, making them superior in stock price prediction (Moghar & Hamiche, 2020; Lin et al., 2021). Compared to traditional neural networks, LSTM performs better in financial market prediction (Uvarajan & Usha, 2024; Palash & Dhurvey, 2024). The models of LSTM are efficient in acquiring long-term data dependency and can effectively deal with uncertainty and noise in the financial market to increase the model's adaptability and strength (Md et al., 2023; Zou et al., 2024). TCN can avoid the problem of information loss in cases where Time Series (TS)are involved as compared to the causal convolutional LSTM structure which makes sure that, at the current moment, future data will not be utilized for forecasting (Dai et al., 2022). TCN can better handle nonlinear and nonsmooth data, and has stronger feature extraction capability and higher prediction accuracy than traditional machine learning models (SVR, RF) (Liu et al., 2023; Barhoumi et al., 2024). Compared to traditional DL models (LSTM, GRU), TCN excels in handling multidimensional features and TSD (Yao et al., 2023). Compared to using LSTM or TCN models alone, the TCN-LSTM model exhibits higher robustness and accuracy in handling nonlinear and unstable PV power data (Limouni et al., 2023). Combining TCN as well as LSTM's advantages, our model can efficiently process local features and global context information to provide more comprehensive prediction results (Ren et al., 2023; Huo et al., 2022). Despite the advancements in DL, there remain significant challenges, even know the role of each hyperparameter in the model, it is still hard to figure out the exact hyperparameter which requires adjustment and how to make the adjustment to acquire the optimal model (Zhao et al., 2020). GA is outstanding in solving the tuning problem. When constructing the LSTM neural network algorithm to predict the closing price, the GA can be utilized to ensure the accuracy of the model prediction (Jia, 2022). Compared to other models, GA-LSTM shows higher stability and prediction accuracy in both the tasks of predicting short-term and long-term prices of stocks (Thakkar & Chaudhari, 2022).

To deal with the mentioned challenges, our study optimizes and develops a new hybrid model to predict stock indexes that consider 7 technical indicators (Prabu et al., 2018). The proposed GA-TCN-LSTM model focuses on utilizing each component's strength: TCN uses extended causal convolution for feature extraction and can handle hierarchical temporal and spatial information of the input data. It captures long-term dependencies as well as local features in the TS. The global information of the TS is kept and passed by LSTM through gating units. After building the TCN-LSTM model, its hyperparameters are optimized using GA. By addressing the limitations of existing methods and proposing a robust hybrid model, our research has contributed in the following ways:

- To develop an effective hybrid model for time-series forecasting.
- For evaluating the performance of the model that has been proposed in comparison to conventional state-of-the-art methods.
- To analyze the model's applicability in different real-world environments.

The remaining parts of the paper are: Related work in the stock market forecasting sector, as well as other hybrid models, is depicted in section 2. The integration of all components and the algorithmic design is depicted in section 3. Section 4 describes the experimental implementation details. In section 5, outcomes obtained from experiments are represented as well as the performance of the model that has been proposed is discussed. In Section 6, this paper outlines as well as concludes possible research options in the future.

2 Related Work

The focus of this review is to enhance the mentioned technologies in today's stock market. LSTM model results are better than the ARIMA model as well as the conventional linear regression model, and when there is an increase in training epochs, the accuracy of prediction also increases (Basnet et al., 2019). An innovative optimization approach, proposed (Prabu et al., 2018) for predicting stocks based on the LSTM model is more efficient than other approaches of DL (Deep Learning) as well as ML (Machine Learning). Gao & Zhang, (2023) used a hybrid neural network model VMD-LSTM to investigate the effect of the sentiments of investors on volatility in price in the Chinese capital market, and forecasting accuracy is better than both the GARCH and VAR models. Findings of another study showed that the CEEMDAN-LSTM model performed more efficiently than the traditional ARIMA model and the single LSTM model concerning stability as well as its ability to make predictions (Lin et al., 2021). Yang et al., (2024) compared the combination of LSTM with ARIMA, SVM, MLP and other models. It is evident from the results of many cases that hybrid models perform much more efficiently than single models, especially in cases where nonlinear time series and complex data are involved. Using a cascade structure, model CLSTM-PPO outperforms the PPO, MLP, LSTM, and LightGBM models (Zou et al., 2024). Gülmez, (2023) used an artificial rabbit optimization algorithm for hyperparameter optimization of an LSTM to enhance its accuracy in predicting information. Successfully found the potential of combining meta-heuristic algorithms to optimize DL models. TCN is better than the GARCH model when using it to predict volatility in the stock market. In the value-at-risk prediction, TCN performs more efficiently than GRU and LSTM in several metrics (Zhang et al., 2022). Compared with other base models, the combination of multivariate empirical modal decomposition and TCN shows a significant improvement in prediction accuracy. Other research has shown that the TCN-Attention framework outperforms the GARCH family models and the LSTM model in describing the dynamics of the ultra-high frequency stock price movement series, especially in predicting extreme price movements (Malathi et al., 2019). Liu et al., (2023) extracted stock features by TCN and channel-time dual attention module. The results show that novel mechanisms capture the short and long-term dependencies more effectively and enhance the accuracy of prediction. Chen et al., (2024) introduced a DL model which is hybrid in nature i.e.; MKP-TemporalNet, it integrates the properties of attention mechanisms, iTCNs and BiGRUs. Its prediction accuracy significantly outperforms traditional LSTM and GRU. Also, the TCN model shows better performance and accuracy in prediction relative to traditional ARIMA, XGBoost and LightGBM.

Liu et al., (2024) proposed a TCN-LSTM model based on a parallel architecture and combined with a Savitzky- Golay filter to smooth the input wind speed time series. It is shown that the model has improved its performance while maintaining the prediction accuracy. Comparing the prediction performance of CEEMDAN-LSTM- TCN, CEEMDAN-BP, CEEMDAN-LSTM, and CEEMDAN-TCN models, the results show that CEEMDAN-LSTM- TCN is the best (Hu et al., 2022). The R2 score of TCN-LSTM in predicting PM2.5 and PM10 concentrations reached 0.948 and 0.876, respectively, and these were remarkably higher as compared to other hybrid models and single models (Ren et al., 2023). This indicates that TCN-LSTM is more advantageous in cases where time series prediction with multifactor-nonlinear features is involved. The TCN-LSTM was utilized for parameter prediction of the sweet spots' reservoir as well (Huo et al., 2022). The model has higher prediction accuracy and also outperforms the simple TCN model in predicting reservoir evaluation parameters at different depths. Limouni et al., (2023) did a comparison of the CNN-LSTM, TCN-LSTM, LSTM and TCN models' performance in the prediction of PV power at various seasons, and the results depicted that the TCN-LSTM model's RMSE and MAE were lesser than other models in all seasons.

GA has the capability of optimizing LSTM's hyperparameters, and GA-LSTM performs way better than the single LSTM model and other traditional machine learning models in prediction accuracy (Chung & Shin, 2018). Wu et al., (2021) introduced a trading system for stocks that combines LSTM as well as GA, called the LSTMLI framework. It outperforms other trading strategies such as the SSACNN+GA model in terms of returns and improves the effectiveness of the trading strategy. Beniwal et al., (2023) compare OGA-SVR's efficiency with conventional SVR, GS-SVR, LSTM as well as GA-SVR. It is found that GA shows its superior performance in optimizing the multi-parameter decision-making process. Thakkar & Chaudhari, (2024) reviewed the applicability of genetic algorithms in stock market forecasting and found GA can be combined with DL methods to optimize hyper-parameters and further enhance the accuracy of prediction. Results from experiments depict that GA outperforms other meta-heuristic algorithms (particle swarm optimization, differential evolution, etc.) in optimizing the parameters of stock price prediction models. Baek, (2024) used CNN for extracting features from LSTM as well as stock to show the history of the data process of input time series in a long-term aspect, combined with GA's use in searching for optimal parameters. Results from the experiments prove that GA-based CNN-LSTM's accuracy in prediction is higher compared to LSTM, single CNN as well as CNN-LSTM.

3 Algorithmic Design

3.1. Temporal Convolutional Networks

TCN is a DL model used to process TSD. This research reveals that it contains dilation, casual convolutional layers, and residual blocks. The model structure includes feature extraction and temporal information extraction modules that utilize causal convolution to maintain the temporal order of the data. Bai et al., (2018) detailed TCN to handle temporal data features with several key properties:

(1) Causal Convolutions: ensure that the convolution operation only relies on previous and current moments, which prevents future data leakage into current prediction scenarios. For a filter $F = (f_1, f_2, \dots, f_k)$ as well as a convolutional kernel size K, based on $x_1, x_{2,\dots}, x_t$to predict $y_1, y_{2\cdot,\dots}, y_t$ the definition of causal convolution at x_t is depicted in Equation (1):

$$(F * X)_{(x_t)} = \sum_{k=1}^{K} f_k x_{t-K+k} \qquad (1)$$

(2) Dilated Convolutions: a convolution operation that consists of an extra rate of hypermeter dilation, which is several intervals in Kernel. The receptive field is increased by inserting a certain number of dilation rates between the convolution kernels. In equation (2), for a sequence X and dilation rate d, Dilated convolution at x_t is defined as:

$$(F *_d X)_{(x_t)} = \sum_{k=1}^{K} f_k x_{t-(K-k)d} \qquad (2)$$

To obtain causality and to make the input and output sequences equal in size, left zero padding is added to the TCN layer. The number of left zero padding p entries required for the layer is shown in Equation (3).

$$p = (K - 1) * d \qquad (3)$$

The dilation convolution allows the network to receive long histories by exponentially increasing the receptive field by adding a dilation factor d to $x_t - d$. The receptive field is represented in Equation (4) which is as follows:

$$R_{field} = 1 + (K-1) * \sum_{m=1}^{n} d_n \qquad (4)$$

(3) Residual Connections: add a skip connection between the convolutional layers. The structure in Figure 1 helps mitigate the problem of vanishing gradients in deep learning, making network training more stable. Allowing it to better capture complex temporal dependencies.

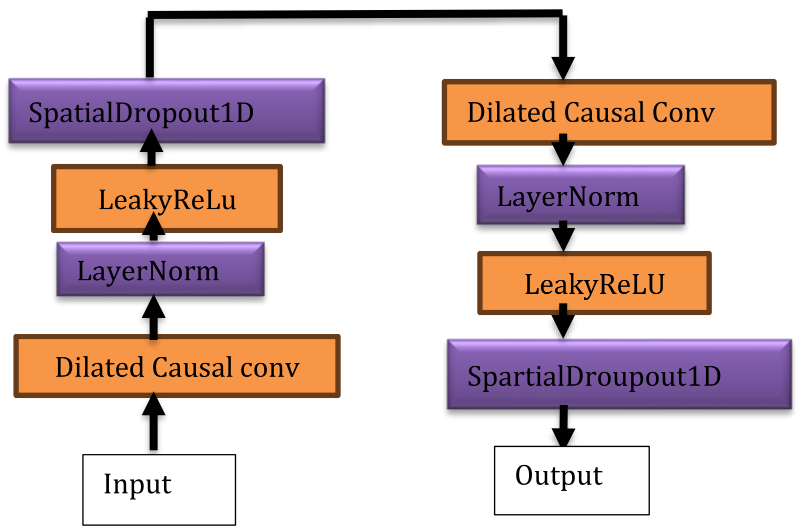

Figure 1: Residual Block

(4) Stacked Layers: increase the model's depth by stacking multiple convolutional layers to increase the model's receptive field and depth. As represented in Figure 2, by stacking convolutional layers, TCN can acquire long-term dependencies as well as complex patterns, in TSD.

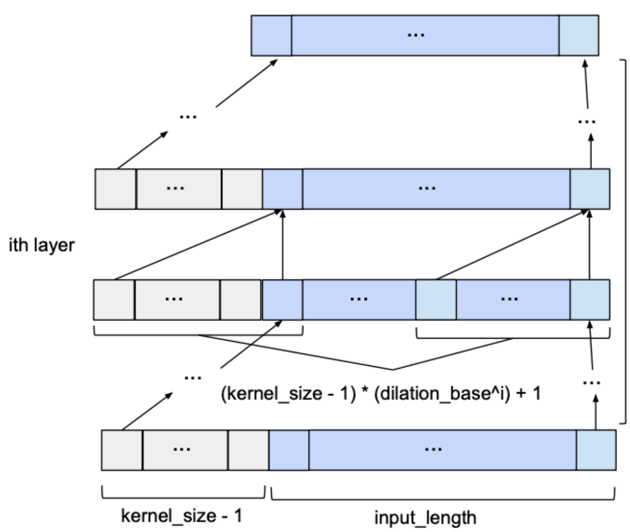

Figure 2: Stacked Layers

3.2. Long Short-Term Memory

Hochreiter & Schmidhuber, (1997) proposed LSTM, which is RNN's variant. Its Memory cell is shown in Figure 3, the operations are processed in each cell of the LSTM model which is responsible for saving important information. Stock prices are affected by past price changes and short- and long-term fluctuations should be considered at the same time. In Figure 3, The structure of LSTM can learn adaptively to better process stock data through its memory cells and control the importance of long-term and short-term information through all gates.

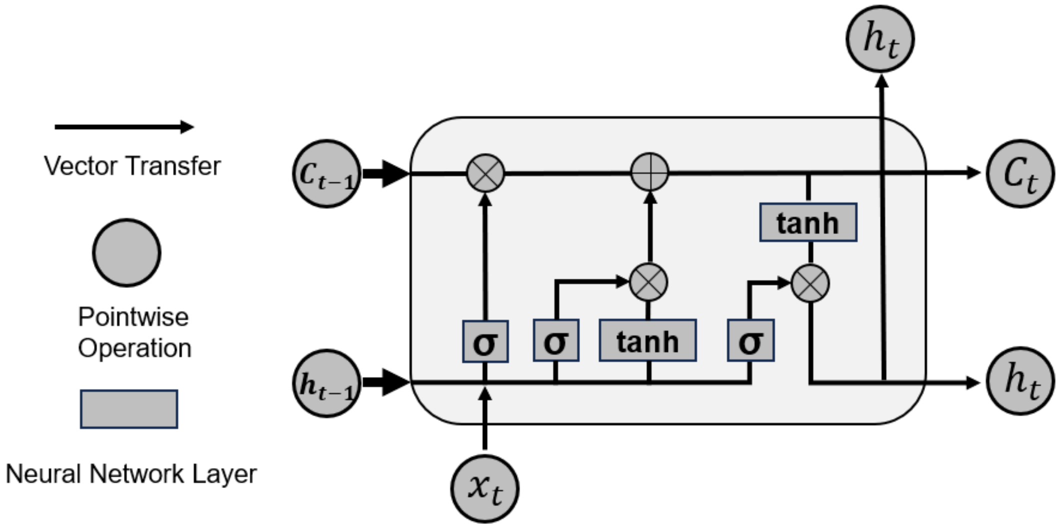

Figure 3: LSTM cell

- Forget Gate (f_t): Decides the discard of information from cell state through Equation (5):

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \qquad (5)$$

- Input Gate (i_t): Decides on where the new information will be stored in the cell state, with Equation (6) and Equation (7)

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \qquad (6)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \qquad (7)$$

- Cell State (C_t): Cell-state is updated, C_{t-1} is updated to C_t as depicted in Equation (8)

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \qquad (8)$$

- Output Gate (O_t): Output value is decided by implementing Equation (9) and Equation (10):

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \qquad (9)$$

$$h_t = o_t * \tanh(C_t) \qquad (10)$$

3.3. Genetic Algorithm

It is a search algorithm that simulates nature's biological evolution process to solve complex problems of optimization. It aims to simulate mutation, crossover and natural selection processes for figuring out an optimal solution for the problems by iterating continuously (Holland, 1992). Algorithm 1 shows a detailed depiction of the use of GA in this research.

Algorithm 1 Genetic Algorithm

1: Initialization of population: Generate a certain number of individuals i.e., solutions in a random manner to create an initial population

2: repeat

3: Evaluate fitness: Calculate the value of fitness for each individual, the objective function to be optimized is the fitness function.

4: Selection: Concerning their value of fitness, select the best individuals typically using the tournament selection method.

5: Crossover: Generate new individuals (offspring) by exchanging part of the genes of a pair of individuals

6: Mutation: Change a portion of the new individual's genes randomly to increase diversity in the population.

7: Population renewal: Replace part or all of the population with offspring to create a new generation in the population.

8: until the maximum number of iterations is reached

3.4 GA-TCN-LSTM Model

As depicted in Figure 4, the pre-portion of the model is the TCN and its main function is processing the time series data taken as input. Local features at different time steps are captured by 1D convolution operations and dilated convolution, with residuals connected within the module. Various dilation rates aid the model in capturing features at various scales of time. The model's post-part is the LSTM module which mainly performs the function of processing the extracted features with TCN because of its structure. LSTM can better understand and process sequential information in the input, which allows it to maintain and update its state. It can adaptively learn and adjust the parameters of the model, which enhances the performance of prediction and the model's generalization ability.

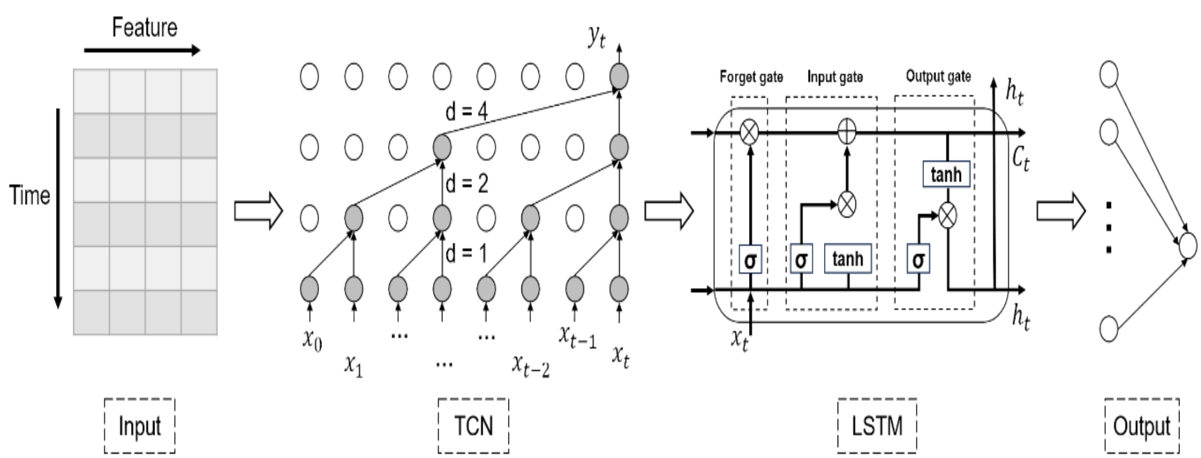

Figure 4: TCN-LSTM Connection

The association of LSTM and TCN helps the model fully utilise the strengths of both capturing short-term features quickly while remembering and processing long-term dependencies, thus improving the prediction performance. After the model's construction, GA is utilized for finding the best parameter combination. The GA optimization process involves initializing the population of potential solutions by randomly generating a set of individuals represented by a binary string. Figure 5 depicts the entire structure of the model that has been proposed.

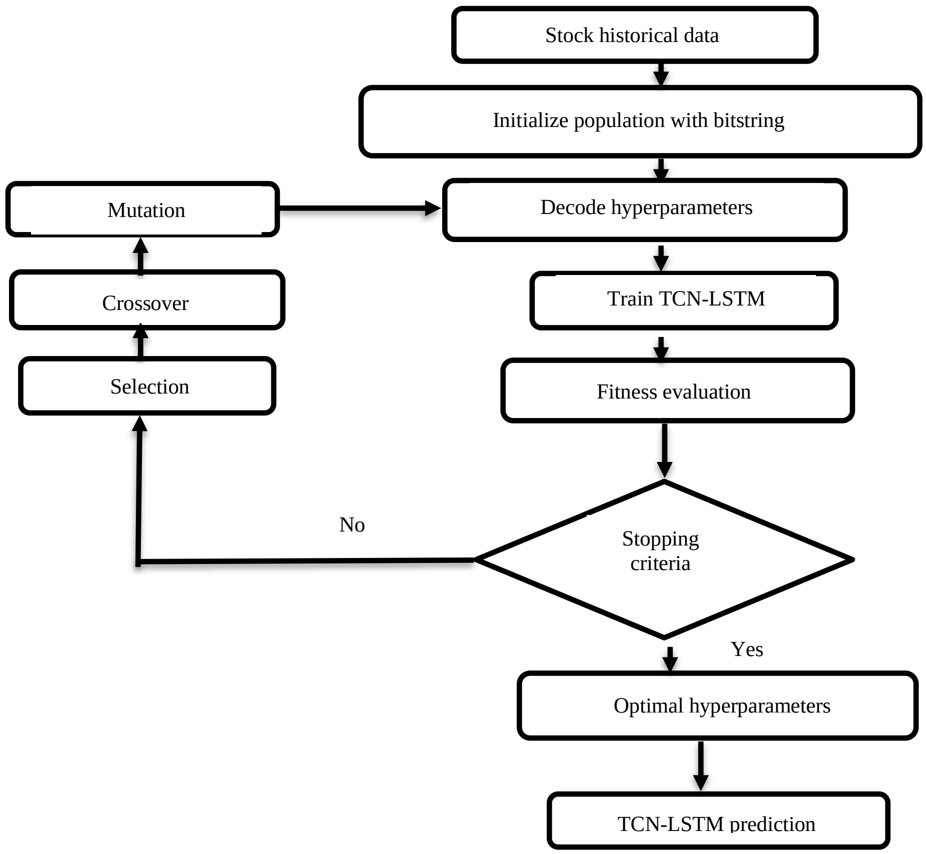

Figure 5: GA-TCN-LSTM Model Prediction Flowchart

The selection of hyperparameters is a critical aspect of model performance and optimization. In this study, the hyperparameters for the TCN-LSTM and GA were carefully chosen, drawing on insights from previous research and validated through extensive experimentation. The specific values and settings, which are pivotal to the capability of the model to learn patterns and achieve convergence, are detailed in Table 1.

Table 1: Hyperparameters Settings

Section	**Hyperparameters**	**Value**
TCN-LSTM	LSTM activation function	Tanh
	TCN activation function	ReLU
	Optimizer	RMSprop
	Fitness function	Huber loss
	Learning rate	0.0001
GA	Dropout rate	0.3
	Number of iterations	20
	Population size	20
	Chromosome length	16
	Crossover rate	0.9
	Mutation rate	0.1
	Tournament size	2

4 Experimental Implementation

To evaluate the superiority and efficiency of the algorithm that has been proposed for predicting prices of stocks, experiments were done on PCs which have Windows 11 as their operating system, GPU (NVIDIA GeForce RTX 3050 4GB) and CPU (13th Gen Intel Core i7-13700H). All the code developed in this work is built from scratch using Python (3.8).

4.1. Data

The performance of 500 large public trading US companies which represent a wide industry spectrum, is tracked by the S&P 500. It is widely regarded as an important indicator of the direction and overall health of the economy of the U.S. economy and financial markets. Using Yahoo Finance's data for U.S. S&P 500 Index from February 20, 2018, to January 1, 2024 (a 6-year period). Table 2 calculated 7 popular indicators from the widely used technical indicators.

Table 2: Technical Indicators Descriptions and Formula

Indicator	Description	Formula
EMA	Gives more weight to recent prices in an attempt to make it more responsive to new information.	$\boldsymbol{P_t \cdot \alpha + EMA_{t-1} \cdot (1-\alpha)}$ $\boldsymbol{\alpha = \frac{2}{N+1}}$
RSI	A momentum oscillator that measures the speed and change of price movements, typically used to identify overbought or oversold conditions.	$\boldsymbol{100 - \frac{100}{1+RS}}$
MACD	A trend-following momentum indicator that shows the relationship between two moving averages of a stock's price.	$\boldsymbol{EMA_{12} - EMA_{26}}$
ATR	Reflects the average volatility of market prices over a given time period.	$\boldsymbol{\frac{1}{n}\sum_{i=1}^{n} TR_i}$
OBV	A momentum indicator that uses volume flow to predict changes in stock price.	$\boldsymbol{OBA_{prev} - Vloume}$
STOCH K%	A momentum indicator comparing a particular closing price of a security to a range of its prices.	$\boldsymbol{\frac{(C - L_{14})}{(H_{14} - L_{14})} \times 100}$
William's R%	A momentum indicator that measures overbought and oversold levels.	$\boldsymbol{\frac{(H_{14} - C)}{(H_{14} - L_{14})} \times (-100)}$

Note: Price at time t is P_t, smoothing factor is α, RS is the relative strength, EMA_{26} and EMA_{12} are the exponential moving averages for 26-days and 12-days, n is the period length, TR_i is the true range, the close price is C, H_{14} and L_{14} are the highest and lowest price over the last fourteen periods.

Figure 6 illustrates the correlation between technical indicators and closing prices for a sample of U.S. market stocks over 6 years. In the scatterplot, correlation coefficient is denoted by r. It is evident that r's absolute value will always be lower than 0.2 for all indicators except EMA and OBV, which means a non-linear relationship exists amongst the closing price and most indicators. This shows that the prediction of closing price of sample stocks with the help of TCN-LSTM is reasonable.

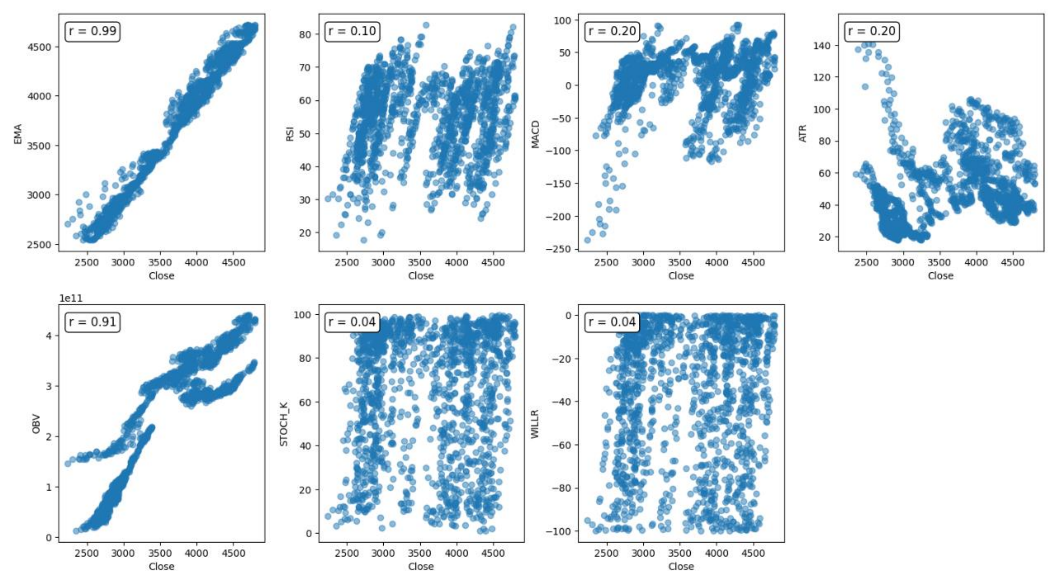

Figure 6: Correlation Between Close Price and Technical Indicator

Then the data is normalized to eliminate the impact of the magnitude scale. By calculating a feature column's maximum as well as minimum values, the MinMaxScaler achieves normalization so that it falls into a specific interval [0,1].

$$\mathbf{X}_{\text{scaled}} = \frac{\mathbf{X} - \mathbf{X}_{\min}}{\mathbf{X}_{\max} - \mathbf{X}_{\min}} \qquad (11)$$

In Equation (11), $\mathbf{X}_{\max}$ and $\mathbf{X}_{\min}$ are the respective maximum and minimum values of the feature, the feature's original value is denoted by $\mathbf{X}$ and scaled value by $\mathbf{X}_{\text{scaled}}$.

4.2. Sliding Window

80% of data gathered over six years was segregated into a training set; the other 20% was utilized as a testing set. In time series data, previous values of selected independent variables are utilized as input features for a specified number of time steps, so the sliding window approach is applied subsequently. This approach is used to dissect the stock index data into a set of overlapping windows. Fixed-size windows are moved through the time series in step 1 as shown in Figure 7.

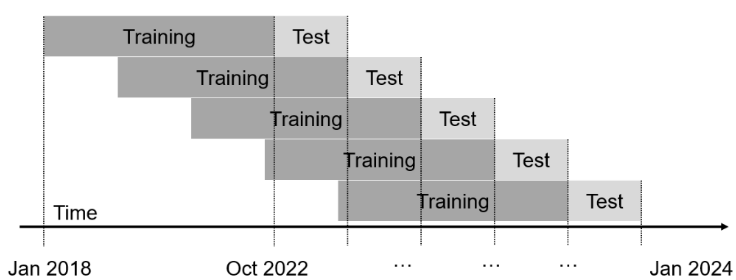

Figure 7: Schematic Representation of the Sliding Window

In the training of the model, use the Early Stopping strategy to avoid overfitting and accelerating optimization. This approach checks the validation loss and if there is no improvement in it even after 20 epochs, the training process is ceased. At this point, the model weights that show the best performance over a validation set are restored. The search spaces of the TCN-LSTM models used in our experiments are also shown in Table 3.

Table 3: Hyperparameters Search Range

Model	Hyperparameter	Range
	Filters in layers 1, 2, 3	2 - 16
	Layers	1 - 5
TCN-LSTM	Neurons	8 - 128
	Batch size	8 - 256
	Window size	1 - 20

4.3. Evaluation Metrics

This research selected four indicators to conduct a performance evaluation of the prediction model. These include R2 which is a measure of model fit. MAE can reflect the error's reality in the value that has been predicted. Instead of MAPE, Weighted MAPE was chosen because it exhibits a performance that is more consistent with the other metrics used to evaluate performance. As well as the RMSE is utilized for measuring the deviation from actual to observed value and the ratio of actual value and error is also taken into consideration. The calculation of these metrics is done with the following formulas:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2} \tag{12}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i| \tag{13}$$

$$\mathrm{wMAPE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{\sum_{i=1}^{n} |y_i|} \times 100 \tag{14}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{15}$$

In Equations (12), (13), (14) and (15), the number of observations is denoted by n, the observed value and the predicted value for i^{th}Observation is denoted by y_i and $\hat{y}_i$ respectively, and observed values' mean is denoted by $\bar{y}$.

In addition to the model performance metrics, we also consider backtesting to assess how good the model is in terms of profitability. Except Accumulative and Annualized Return, Atilgan et al., (2013) also use Sharpe Ratio, Calmar Ratio, Maximum Drawdown in Table 4 to measure stock risk.

Table 4: Profitability Metrics

Metric	Description	Formula
Acc. R (%)	Measures the total percentage increase or decrease in an investment's value over a specific period.	$\left(\frac{P_{\text{end}}}{P_{\text{start}}} - 1\right) \times 100\%$
Ann. R (%)	Converts total returns into an average yearly return, making it easier to compare the performance of investments held for different periods.	$\left(\frac{P_{\text{end}}}{P_{\text{start}}}\right)^{\frac{1}{n}} - 1 \times 100\%$
S. R.	Evaluates the excess return per unit of risk (volatility) compared to a risk-free investment.	$\frac{E[R_p - R_f]}{\sigma_p}$
C. R.	Assesses the risk-adjusted return by comparing annualized returns to the maximum drawdown.	$\frac{\text{Annualized Return}}{\text{Maximum Drawdown}}$
MDD(%)	Indicates the greatest loss from a peak to a trough during the investment period.	$\max\left(\frac{V_{Peak} - V_{Trough}}{V_{Peak}}\right) \times 100\%$

Note: Return of the stock is denoted by R_p, number of years by n, Risk-free rate (default 1%) by R_f, Standard deviation of excess return of stock by σ_p, V_{Peak} and V_{Trough} is the highest and lowest value reached.

5 Experiment Results

5.1. Experiment 1: Effectiveness of GA-TCN-LSTM

The first experiment tests the efficiency when GA and TCN-LSTM are combined. It is evident from Figure 8 that TCN-LSTM with optimized parameters by GA algorithm can obtain good fit at both training and testing set.

Figure 8: GA-TCN-LSTM prediction v.s. actual

To assess its performance, we compare it to the performance of a set of reference models optimized using a grid search approach on the U.S. stock market. The reference models include TCN, LSTM, GRU, and RNN, and the numbers in Table 5 indicate the final settings that produce the best results for each model. The genetic algorithm approach is only applied to the TCN-LSTM models as it is designed to improve their performance and compare their effectiveness with the grid search on the reference models. To be fair, all competing algorithms were stopped after 500 epochs.

Table 5: Model Hyperparameters Selection on S&P 500

Method	Model	Hyperparameter	Value
Grid Search	LSTM	Layers	1
		Neurons	40
		Batch size	32
		Window size	5
	TCN	Filters of layer 1	9
		Filters of layer 2	3
		Filters of layer 3	3
		Batch size	32
		Window size	1

		Layers	1
		Neurons	40
		Batch size	32
		Window size	20
	GRU	Layers	1
		Neurons	50
		Batch size	32
		Window size	1
Genetic Algorithm	RNN	Filters of layer 1	58
		Filters of layer 2	84
		Filters of layer 3	23
	TCN-LSTM	Layers	1
		Neurons	197
		Batch size	13
		Window size	54

It is evident from Table 6 that GA-TCN-LSTM significantly outperforms all other models across all metrics. It achieves the lowest RMSE of 16.53, which is substantially better than the next best model, TCN, with an RMSE of 40.88. Similarly, the GA-TCN-LSTM model exhibits the lowest wMAPE (0.32%) and MAE (13.34), indicating its superior accuracy. The R2 score of 0.996 further confirms that this model explains nearly all the variance in the data, highlighting its robustness. In contrast, the LSTM shows the poorest performance with maximum RMSE (57.68), wMAPE (1.08%), and MAE (45.60), and the lowest R2 score (0.948), showing that it is less effective in this context. Overall, amongst the values that have been evaluated, the most reliable and accurate model is the GA-TCN-LSTM.

Table 6: Model Performance Comparison on S&P 500

Model	RMSE	wMAPE(%)	MAE	R2 score
TCN	40.88	0.76	32.18	0.974
LSTM	57.68	1.08	45.60	0.948
GRU	42.43	0.79	33.36	0.972
RNN	43.45	0.81	34.37	0.971
GA-TCN-LSTM	16.53	0.32	13.34	0.996

Figure 9 shows a more visual depiction of stock index movements predicted by the different models compared to the actual S&P 500 (solid black line). GA-TCN-LSTM (red line) stands out as the most accurate model, closely tracking the actual S&P 500 values throughout the entire period. Its performance shows minimal deviation from the black line, indicating strong predictive power and robustness in both rising and falling market trends. TCN (orange line) and LSTM (blue dashed line) also demonstrate relatively strong performance, particularly during periods of smoother market movement. However, they occasionally exhibit larger deviations from the actual index values, especially during periods of high volatility. RNN (purple dashed line) and GRU (green dashed line) tend to lag slightly behind in performance, displaying more noticeable deviations from the actual values, particularly during significant market fluctuations. Their predictions appear less stable compared to GA-TCN-LSTM. In summary, GA-TCN-LSTM emerges as the superior model for this time series prediction task, outperforming the other models in having a more efficient consistency and accuracy across various conditions of market. The hybrid model's ability in effectively capturing complex temporal dependencies likely contributes to its optimal performance.

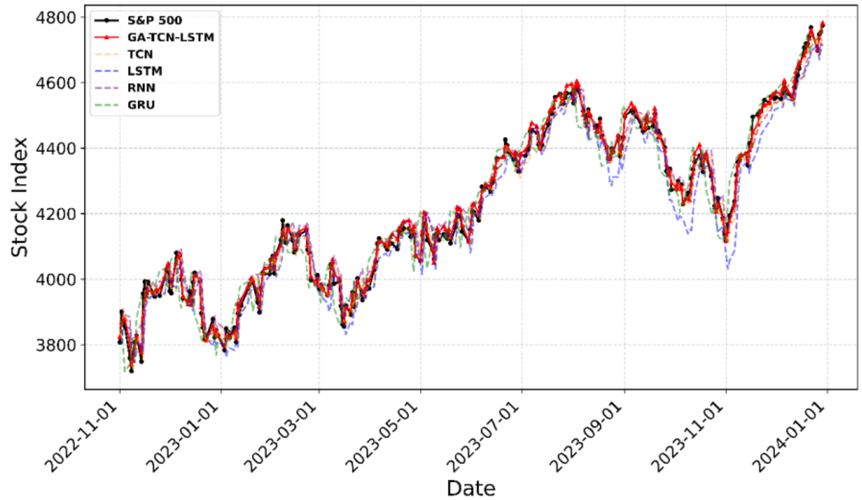

Figure 9: Model Trend Comparison

5.2. Experiment 2: Profitability of the GA-TCN-LSTM

S&P 500 index's accumulative return is utilized as a typical benchmark. The trend depicted in Figure 10 visualizes the portfolios' average cumulative return acquired through deep learning in U.S market. The strategy of buy-and-hold is the benchmark, it also represents S&P 500 index gain. It can be observed the accumulative returns of the hybrid GA-TCN-LSTM are constantly higher than the accumulative returns of the hybrid other algorithms and much higher than the returns of the generic benchmark. Moreover, the advantage of GA-TCN-LSTM over the benchmark is expanding dramatically as the vintage increases. Hence, it proves that GA offers more efficient parameter settings than other algorithms, for TCN-LSTM model.

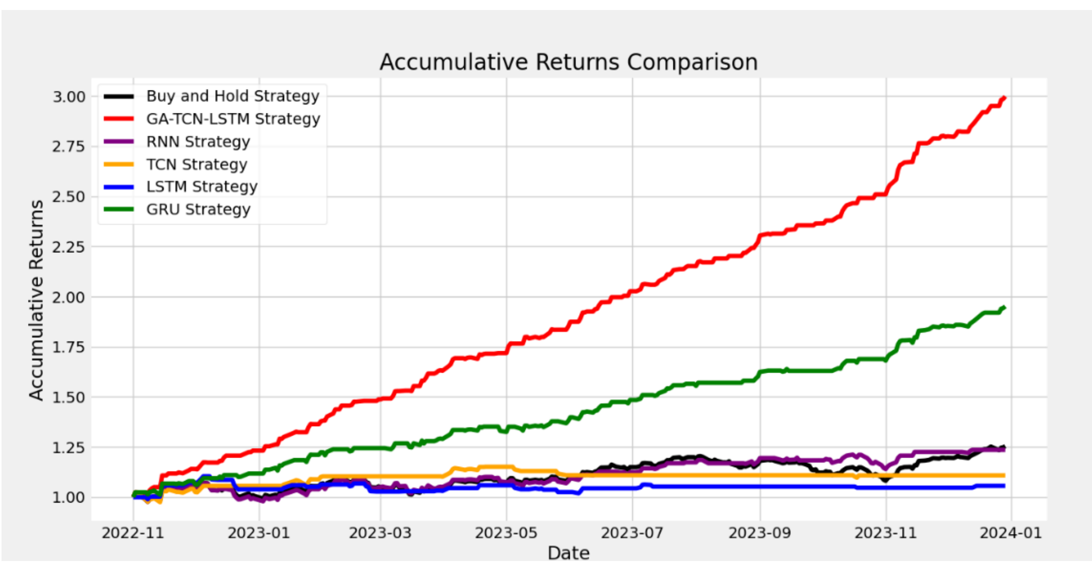

Figure 10: Accumulative Returns Comparison

A detailed comparison of the portfolio's overall performance acquired from a neural network model and GA-TCN-LSTM algorithm over a six-year period in terms of Annualized return, Sharpe ratio, Accumulative return, Maximum Drawdown Rate and Calmar ratio is depicted in Table 7.

Table 7: Profitability Performance Comparison

Strategy	B & H	GA-TCN-LSTM	TCN	LSTM	RNN	GRU
Acc.R	0.25	2.00	0.11	0.06	0.24	0.95
An.R	0.22	1.58	0.09	0.05	0.20	0.78
S.R.	1.30	9.24	0.93	0.48	1.30	4.93
C.R.	2.10	344.87	1.93	0.63	2.34	37.56
MDD(%)	10.30	0.46	4.82	7.69	8.57%	2.08

GA-TCN-LSTM model outperforms all other strategies across all metrics. The comparison across different strategies highlights the clear superiority of itself. It achieves an accumulated return of 2.00 and also the highest annualized return of 1.58, significantly outperforming all other strategies, with the next best being GRU at only 0.95 and 0.78. Notably, the Sharpe ratio of GA-TCN-LSTM stands out at 9.24. The most striking difference is seen in the accumulative return, where GA-TCN-LSTM achieves an astonishing 344.87, vastly outperforming all other models. Additionally, the maximum drawdown is the lowest at 0.46%, indicating the highest level of risk management among the strategies. Other models like GRU perform well in terms of Sharpe ratio and Calmar ratio. Traditional methods like buy and hold show mediocre performance in comparison, and models like LSTM and RNN perform poorly overall. These results collectively underscore GA-TCN-LSTM's important benefit in delivering superior returns while minimizing risk.

5.3. Experiment 3: Application in Different Stock Markets

In this experiment, GA-TCN-LSTM is applied to 5 different stock markets to investigate the robustness of the model. Including the U.S. (NASDAQ), China (CSI 300), Hong Kong (HS 50), and Japan (Nikkei 225). Tables 8 and 9 depict predictive accuracy of the model and profitability vary significantly across different markets.

In terms of predictive performance, the model performs best in the S&P 500 market, with all metrics indicating excellent prediction accuracy in this market. In contrast, the RMSE values in the Nikkei 225 and NASDAQ markets are relatively higher, at 331.99 and 154.19, respectively, though the R2 scores remain reasonably strong at 0.983 and 0.985, suggesting the model can still capture trends effectively in these markets.

Table 8: Effectiveness in Different Stock Markets

Stock Market	RMSE	wMAPE （%）	MAE	R2 score
S&P 500	16.53	0.32	13.34	0.996
NASDAQ	154.19	0.97	122.30	0.985
CSI 300	33.34	0.71	26.19	0.968
HS 50	148.33	0.62	117.35	0.991
Nikkei 225	331.99	0.87	263.53	0.983

Regarding profitability, the GA-TCN-LSTM model performs particularly well in the HS 50 market, achieving a cumulative return of 3.75, an annualized return of 2.91, and a Sharpe ratio of 7.34, indicating significant returns with favourable risk-adjusted performance in this market. However, in the NASDAQ market, despite a lower cumulative return of 0.26 and a Sharpe ratio of 1.34, the maximum drawdown is 9.84%, which is higher than in other markets, suggesting greater volatility and higher risk exposure for the model in this market.

Table 9: Profitability in Different Stock Markets

Stock Market	S&P 500	NASDAQ	CSI 300	HS 50	Nikkei 225
Acc.R	2.00	0.26	0.19	3.75	0.31
An.R	1.58	0.22	0.12	2.91	0.27
S.R.	9.24	1.34	1.19	7.34	2.23
C.R.	344.87	2.19	1.74	93.88	5.07
MDD(%)	0.46	9.84	6.88	3.10	5.36

Note: The * indicates values that are lesser than the strategy of buy-and-hold.

In summary, the GA-TCN-LSTM model demonstrates a robust ability to adapt to different stock markets, showing its flexibility and effectiveness across various economic environments. While its predictive accuracy and profitability are particularly strong in the S&P 500 and HS 50 markets, making it suitable for long-term investments, the model also showcases a remarkable ability to capture trends in more volatile markets such as NASDAQ and Nikkei 225. Despite the relatively higher RMSE values in these markets, the R2 scores indicate that the model still maintains strong predictive performance. However, the increased market risk and volatility in NASDAQ and Nikkei 225 suggest that while the GA-TCN-LSTM model is resilient, a more cautious investment strategy may be necessary to mitigate the effects of heightened drawdowns and market fluctuations. This robustness highlights the model's potential to be a versatile tool in stock market prediction across varying conditions.

6 Conclusion and Future Work

Experimental results across multiple international stock markets demonstrate the high-performance efficiency of GA- TCN-LSTM compared to the conventionally used TCN, GRU, LSTM as well as RNN models. The GA-TCN-LSTM performs more efficiently than these models in offering higher accuracy of prediction, measured by RMSE, MAE, and R^2, and in terms of profitability, measured by parameters like Sharpe Ratio, Annual Returns, and Maximum Drawdown. The results prove that drawbacks in the existing approaches are addressed effectively by the hybrid model such as the inability to capture dependencies which are long-term as well as the challenge of handling nonlinear relationships in financial time series data. Incorporating GA helps optimize the parameters, improving the model's overall performance. Furthermore, TCN and LSTM components work synergistically to provide a robust framework for time-series forecasting. Its adaptability to different market conditions makes it a versatile and reliable model for stock market forecasting.

Future work can explore other optimization techniques alongside GA could provide additional insights and potentially enhance the model's performance. Integrating other relevant sources of data such as financial news, social media sentiment, and macroeconomic indicators, could provide a more comprehensive view of the factors influencing stock prices. Extending the model so that it can deal with multi-variate time series data as well as testing it on larger, more complex datasets could further demonstrate its applicability and scalability.

Statement of Availability of Data

Availability of data will be ensured whenever it is requested.

Statement of Credit Authorship Contribution

Competing Interest Declaration

It is declared by the authors that they do not have personal relationships or competing financial interests which could influence the work reported in the paper.

Acknowledgements

The authors would like to thank anonymous reviewers and editors for their valuable comments.

Funding

Our research did not receive any specific funding.

References

[1] Ahangar, R. G., Yahyazadehfar, M., & Pournaghshband, H. (2010). The comparison of methods artificial neural network with linear regression using specific variables for prediction stock price in Tehran stock exchange. https://doi.org/10.48550/arXiv.1003.1457

[2] Atilgan, Y., Bali, T. G., & Demirtas, K. O. (2013). The performance of hedge fund indices. *Borsa Istanbul Review*, *13*(3), 30-52. https://doi.org/10.1016/j.bir.2013.10.007

[3] Baek, H. (2024). A CNN-LSTM stock prediction model based on genetic algorithm optimization. *Asia-Pacific Financial Markets*, *31*(2), 205-220. https://doi.org/10.1007/s10690-023-09412-z

[4] Bai, S., Kolter, J. Z., & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. https://doi.org/10.48550/arXiv.1803.01271

[5] Barhoumi, E. M., Charabi, Y., & Farhani, S. (2024). Detailed guide to machine learning techniques in signal processing. *Progress in Electronics and Communication Engineering*, *2*(1), 39–47.

[6] Basnet, R. B., Shash, R., Johnson, C., Walgren, L., & Doleck, T. (2019). Towards Detecting and Classifying Network Intrusion Traffic Using Deep Learning Frameworks. *Journal of Internet Services and Information Security*, *9*(4), 1-17.

[7] Beniwal, M., Singh, A., & Kumar, N. (2023). Forecasting long-term stock prices of global indices: A forward-validating Genetic Algorithm optimization approach for Support Vector Regression. *Applied Soft Computing*, *145*, 110566. https://doi.org/10.1016/j.asoc.2023.110566

[8] Chen, W., An, N., Jiang, M., & Jia, L. (2024). An improved deep temporal convolutional network for new energy stock index prediction. *Information Sciences*, *682*, 121244. https://doi.org/10.1016/j.ins.2024.121244

[9] Chung, H., & Shin, K. S. (2018). Genetic algorithm-optimized long short-term memory network for stock market prediction. *Sustainability*, *10*(10), 3765. https://doi.org/10.3390/su10103765

[10] Dai, W., An, Y., & Long, W. (2022). Price change prediction of ultra high frequency financial data based on temporal convolutional network. *Procedia Computer Science*, *199*, 1177-1183. https://doi.org/10.1016/j.procs.2022.01.149

[11] Dash, R., & Dash, P. K. (2016). A hybrid stock trading framework integrating technical analysis with machine learning techniques. *The Journal of Finance and Data Science*, *2*(1), 42-57. https://doi.org/10.1016/j.jfds.2016.03.002

[12] Gao, Z., & Zhang, J. (2023). The fluctuation correlation between investor sentiment and stock index using VMD-LSTM: Evidence from China stock market. *The North American Journal of Economics and Finance*, *66*, 101915. https://doi.org/10.1016/j.najef.2023.101915

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

[14] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[15] Holland, J. H. (1992). Genetic algorithms. *Scientific american*, *267*(1), 66-73.

[16] Hu, C., Zhao, Y., Jiang, H., Jiang, M., You, F., & Liu, Q. (2022). Prediction of ultra-short-term wind power based on CEEMDAN-LSTM-TCN. *Energy Reports*, *8*, 483-492. https://doi.org/10.1016/j.egyr.2022.09.171

[17] Hu, Y., Liu, K., Zhang, X., Su, L., Ngai, E. W. T., & Liu, M. (2015). Application of evolutionary computation for rule discovery in stock algorithmic trading: A literature review. *Applied Soft Computing*, *36*, 534-551. https://doi.org/10.1016/j.asoc.2015.07.008

[18] Huo, F., Chen, Y., Ren, W., Dong, H., Yu, T., & Zhang, J. (2022). Prediction of reservoir key parameters in 'sweet spot'on the basis of particle swarm optimization to TCN-LSTM network. *Journal of Petroleum Science and Engineering*, *214*, 110544. https://doi.org/10.1016/j.petrol.2022.110544

[19] Jalaja, V., Sheshadri, T., Arthi, V. K., Thilaga, S., Bamini, J., Basha, S. M., & Kethan, M. (2024). Maximizing Marketing Value: An Empirical Study on the Framework for Assessing AI and ML Integration in Marketing Management. *Indian Journal of Information Sources and Services*, *14*(3), 64-70. https://doi.org/10.51983/ijiss-2024.14.3.09

[20] Jia, H. (2022, April). Based on the LSTM-GA Stock Price Ups and Downs Forecast Model. In *International Conference on Multi-modal Information Analytics* (pp. 117-124). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-05237-8_15

[21] Limouni, T., Yaagoubi, R., Bouziane, K., Guissi, K., & Baali, E. H. (2023). Accurate one step and multistep forecasting of very short-term PV power using LSTM-TCN model. *Renewable Energy*, *205*, 1010-1024. https://doi.org/10.1016/j.renene.2023.01.118

[22] Lin, Y., Yan, Y., Xu, J., Liao, Y., & Ma, F. (2021). Forecasting stock index price using the CEEMDAN-LSTM model. *The North American Journal of Economics and Finance*, *57*, 101421. https://doi.org/10.1016/j.najef.2021.101421

[23] Liu, H., Zhao, T., Wang, S., & Li, X. (2023). A stock rank prediction method combining industry attributes and price data of stocks. *Information Processing & Management*, *60*(4), 103358. https://doi.org/10.1016/j.ipm.2023.103358

[24] Liu, S., Xu, T., Du, X., Zhang, Y., & Wu, J. (2024). A hybrid deep learning model based on parallel architecture TCN-LSTM with Savitzky-Golay filter for wind power prediction. *Energy Conversion and Management*, *302*, 118122. https://doi.org/10.1016/j.enconman.2024.118122

[25] Mabu, S., Obayashi, M., & Kuremoto, T. (2015). Ensemble learning of rule-based evolutionary algorithm using multi-layer perceptron for supporting decisions in stock trading problems. *Applied soft computing*, *36*, 357-367. https://doi.org/10.1016/j.asoc.2015.07.020

[26] Malathi, K., Dhivya, E., Monisha, M., & Pavithra, P. (2019). Preterm birth prognostic prediction using Cross domain data fusion. *International Journal of Communication and Computer Technologies*, *7*(1), 10-13.

[27] Md, A. Q., Kapoor, S., AV, C. J., Sivaraman, A. K., Tee, K. F., Sabireen, H., & Janakiraman, N. (2023). Novel optimization approach for stock price forecasting using multi-layered sequential LSTM. *Applied Soft Computing*, *134*, 109830. https://doi.org/10.1016/j.asoc.2022.109830

[28] Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. *Procedia computer science*, *170*, 1168-1173. https://doi.org/10.1016/j.procs.2020.03.049

[29] Palash, P. S., & Dhurvey, P. (2024). Influence and Prediction of Sintered Aggregate Size Distribution on the Performance of Lightweight Alkali-Activated Concrete. *Archives for Technical Sciences*, 2(31), 49–56. https://doi.org/10.70102/afts.2024.1631.049

[30] Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert systems with applications*, *42*(4), 2162-2172.

[31] Prabu, G. R. K., Murugesan, M., & Ramachandran, G. (2018). Efficient Maximum Power Point Tracking Using Model Predictive Fuzzy Logic Control for Photovoltaic Systems under Dynamic Weather Condition. *International Journal of Advances in Engineering and Emerging Technology*, *9*(3), 104-118.

[32] Ren, Y., Wang, S., & Xia, B. (2023). Deep learning coupled model based on TCN-LSTM for particulate matter concentration prediction. *Atmospheric Pollution Research*, *14*(4), 101703. https://doi.org/10.1016/j.apr.2023.101703

[33] 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. https://doi.org/10.1016/j.asoc.2020.106181

[34] Thakkar, A., & Chaudhari, K. (2022). Information fusion-based genetic algorithm with long short-term memory for stock price and trend prediction. *Applied Soft Computing*, *128*, 109428. https://doi.org/10.1016/j.asoc.2022.109428

[35] Thakkar, A., & Chaudhari, K. (2024). Applicability of genetic algorithms for stock market prediction: A systematic survey of the last decade. *Computer Science Review*, *53*, 100652. https://doi.org/10.1016/j.cosrev.2024.100652

[36] Uvarajan, K. P., & Usha, K. (2024). Implement A System for Crop Selection And Yield Prediction Using Random Forest Algorithm. *International Journal of communication and computer Technologies*, *12*, 21-26. https://doi.org/10.31838/IJCCTS/12.01.02

[37] Wu, J. M. T., Sun, L., Srivastava, G., & Lin, J. C. W. (2021). A Novel Synergetic LSTM-GA Stock Trading Suggestion System in Internet of Things. *Mobile Information Systems*, *2021*(1), 6706345. https://doi.org/10.1155/2021/6706345

[38] Yang, Q., Yu, Y., Dai, D., He, Q., & Lin, Y. (2024). Can hybrid model improve the forecasting performance of stock price index amid COVID-19? Contextual evidence from the MEEMD-LSTM-MLP approach. *The North American Journal of Economics and Finance*, *74*, 102252. https://doi.org/10.1016/j.najef.2024.102252

[39] Yao, Y., Zhang, Z. Y., & Zhao, Y. (2023). Stock index forecasting based on multivariate empirical mode decomposition and temporal convolutional networks. *Applied Soft Computing*, *142*, 110356. https://doi.org/10.1016/j.asoc.2023.110356

[40] Zhang, C. X., Li, J., Huang, X. F., Zhang, J. S., & Huang, H. C. (2022). Forecasting stock volatility and value-at-risk based on temporal convolutional networks. *Expert Systems with Applications*, *207*, 117951. https://doi.org/10.1016/j.eswa.2022.117951

[41] Zhao, S., Zeng, D., Wang, W., Chen, X., Zhang, Z., Xu, F., ... & Liu, X. (2020). Mutation grey wolf elite PSO balanced XGBoost for radar emitter individual identification based on measured signals. *Measurement*, *159*, 107777. https://doi.org/10.1016/j.measurement.2020.107777

[42] Zou, J., Lou, J., Wang, B., & Liu, S. (2024). A novel deep reinforcement learning based automated stock trading system using cascaded lstm networks. *Expert Systems with Applications*, *242*, 122801. https://doi.org/10.1016/j.eswa.2023.122801

Authors Biography

Zeren Shi was born in China, in 2000. He received the Master degree from University of Birmingham in 2023. Now, he is a PhD candidate at Faculty of Management, University Technology Malaysia. His research interest includes Big Data Finance, Financial Engineering and Machine Learning.

Othman Ibrahim received the Ph.D. degree in computation from the University of Manchester Institute of Science and Technology in 2004. Now, he is the Associate Professor in University Technology Malaysia. His research interest includes Information System, Knowledge Management, E-Gov.

Hanini Ilyana Che Hashim received the Ph.D. degree in Transport and Logistics from the Malaysia Institute of Transport in 2021. Now, she is the Senior Lecturer in University Technology Malaysia. Her research interest includes Technology Financing, Corporate Management, Supply Chain.