Estimating Stock Market Prices with Histogram-based Gradient Boosting Regressor: A Case Study on Alphabet Inc

Shigen Li[1]*
Huzhou South the Taihu Lake Economic Management Research Institute / Hangzhou Tianze Security Technology Consulting Firm, Zhejiang, 313000, China / Zhejiang, 310021, China

***Abstract*—One of the most important and common activities mentioned while discussing the financial markets is stock market trading. An investor is constantly searching for methods to estimate future trends to minimize losses and maximize profits due to the unavoidable volatility in stock prices. It is undeniable, nonetheless, that there is currently no mechanism for accurately estimating future market patterns despite numerous approaches being investigated to enhance model performance as much as feasible. Findings indicate notable improvements in accuracy compared to traditional Histogram-based gradient-boosting models. Experiments conducted on historical stock price datasets verify the efficacy of the proposed method. The combined strength of HGBoost and optimization techniques, including Particle Swarm Optimization, Slime Mold Algorithm, and Grey Wolf Optimization, not only increases prediction accuracy but also fortifies the model's ability to adjust to changing market conditions. The results for HGBoost, PSO- HGBoost, SMA-HGBoost, and GWO- HGBoost were 0.964, 0.973, 0.981, and 0.988, in that order. Compared to HGBoost, the result of GWO-HGBoost shows how combining with the optimizer can enhance the output of the given model.**

***Keywords*—Alphabet Inc.; market movement; stock; financial markets; Histogram-based gradient boosting**

I. Introduction

A. Research Background

For investors, forecasting the future price of the stock market is crucial since it lowers the danger of making investment decisions based only on gauging future trends. Because of how volatile the stock market is, it might be difficult to predict future changes. Therefore, appropriate computational techniques are needed to anticipate stock price movement. Many debates on the predictability of the stock market have been gaining traction for decades [1]. Initially, the random walk theory was used to describe how the stock price moved. Later, the Effective Market Hypothesis (EMH) was used to base research on price movements [2], [3]. They believe that past and current values have no bearing on future price movement, and they also think it is impossible to anticipate future stock prices. Alternatively, several studies have attempted to refute the EMH; empirical and observational data have shown that there is some degree of predictive capacity for the stock market. Researchers in the field of stock price forecasting have developed some traditional approaches, such as Autoregressive Moving Average (ARMA), Autoregressive Integrated Moving Average (ARIMA), etc. However, these methods have some limitations because they assume a linear form for the model's structure, which makes them incapable of handling the nonlinear relationships found in time series data [4], [5].

The majority of traditional time series prediction techniques rely on stationary trends, which makes stock price prediction inherently challenging. In addition, the sheer number of factors involved in stock price prediction makes it a difficult problem in and of itself. The market acts like a voting machine in the short run, but it acts like a weighing machine in the long run. Therefore, it is possible to predict market movements for a longer period [6]. The most potent tool is machine learning (ML), which uses a variety of algorithms to improve performance in a given case study [7]. Many people think that ML is very good at finding reliable facts and patterns in the dataset [8]. Some of the machine learning models used for prediction are Decision Trees [9], Random Forests [10], Support Vector Machines [11], Neural Networks [11], Gradient Boosting [12], and Time Series Forecasting [13], [14]. These models succeed at finding underlying trends and patterns in data that can be hard to find with conventional research. This is a very useful skill for seeing trends and openings. However, a few of these models are also flawed. It is possible for prediction models to overfit the training set, resulting in the capture of anomalies and noise instead of true patterns.

As a consequence, the models function well on training data but badly on fresh, untested data. Nonetheless, there are a few strategies and tactics that can be used to raise the models' performance. In several fields, including natural language processing, picture identification, and predictive analytics, machine learning models have become essential. Optimizing hyperparameters is essential to using these models effectively. The selection of hyperparameters, which direct the machine learning algorithms' learning process, has a significant effect on the performance of the model [15]. Optimizing hyperparameters is mostly done to optimize machine learning model performance. A model's capacity to learn from data and generalize to new, unobserved cases is greatly influenced by hyperparameters, including learning rates, regularization strengths, and network designs. Machine learning practitioners try to optimize the model performance by adjusting these hyperparameters [16]. The model presented in this work is Histogram-based gradient boosting (HGBoost); the HGBoost is a machine-learning technique that combines the ideas of histogram-based feature splitting and gradient boosting to solve problems associated with regression. This approach is a modification of the widely used Gradient Boosting Machine (GBM) method [12], [17]. The two main variations of gradient boosting, a machine-learning strategy for prediction, are regression and classification. Unlike earlier methods, this paradigm aims to handle big and complex problems rather than simple and minor ones.

The gradient-boosting method called HGBoost was developed specifically to address regression problems. This method is well known for being fast and efficient in accelerating decision-tree learning. HGBoost does this by discretizing the input variables, which divides additional trees into several values [17]. The optimizers presented in this research to optimize the hyperparameters of the HGBoost model are Particle swarm optimization [18], Slime mold algorithm [19], and grey wolf optimization [20]. PSO Inspired by swarming birds' social behavior, the PSO process is a stochastic search technique. In the search space, each particle in the algorithm represents a possible solution. In addition to being able to hold onto its local and global greatest value, the velocity of the points in the space gives information on how they are moving in that direction [21].

The Physarum polycephalum's behavior and morphological changes during foraging are mostly simulated using the SMA, which was presented by Li et al. [19] in 2020. Weights in SMA were used to model the positive and negative feedback produced during the slime mold's foraging activity, resulting in the formation of three distinct morphological forms of slime mold. Slime mold is a eukaryotic creature that lives in a wet, chilly environment that eats mostly Plasmodium. The organic mass of slime mold searches for food during the active feeding phase envelops it, and secretes digestive enzymes. Its leading edge migrates in sectors and, its trailing end is made up of a web of veins that are linked and permit cytoplasmic movement inside. They may use a range of food sources to create linked venous networks concurrently, according to the characteristics of slime mold. The last optimization used to optimize the hyperparameter of the model, which has the best results, is GWO. The GWO algorithm is a metaheuristic optimization technique that takes its cues from the natural hunting behavior and social hierarchy of grey wolves. GWO is a population-based optimization technique that was created by Seyedali Mirjalili in 2014 and is used to solve challenging optimization issues [20]. It works especially effectively for applications involving combinatorial and continuous optimization. The social dynamics and hunting techniques of a pack of grey wolves serve as the foundation for the GWO algorithm. Alpha, beta, and delta wolves assume leadership positions in these social interactions, which involve leader-follower dynamics.

Real-time data processing using machine learning algorithms enables traders and investors to act swiftly and decisively. This is especially important for the stock market, where news and events can cause values to fluctuate quickly. Machine learning may assist in determining and evaluating the risks connected to various investing possibilities. These models can offer insights into the possible drawbacks of a specific investment by examining historical data and market indicators, assisting investors in making better decisions.

B. *Related Works*

Financial markets have recently used machine learning techniques. Bhalke et al. [22] highlighted the challenges involved in stock market price forecasting by recognizing its intricate and erratic nature. They highlighted the commonality of patterns observed in stock price curves and acknowledged the possibility for machine learning techniques to reduce this complexity by automating forecast processes. Their research article focused on using Long Short-Term Memory (LSTM) networks to estimate future stock market values using daily closing price data. Future stock price predictions and training both made use of LSTM, which is well known for its effectiveness in processing sequential data.

Due to the stock market's mix of high profits and significant dangers, Su et al. [23] underlined how important stock price prediction is for investors, underscoring the stock market's importance in the investing environment. They proposed a method for predicting stock values utilizing the hidden Markov model (HMM) by leveraging advancements in computer technology, such as machine learning and econometric approaches. In other words, they converted the discrete HMM into a continuous HMM to take into consideration the time series continuity of stock price data. Based on the continuous HMM framework, an up-and-down trend model for forecasting was put into practice. This model included methodologies for fluctuation range prediction and extended first-order to second-order continuous HMMs. The model's ability to predict stock prices over six months was demonstrated by using it to duplicate the Hang Seng Index (HSI). The assessment findings showed a good degree of agreement between the actual and projected values, outperforming three benchmark models in terms of RMSE, MAE, and R^2.

The ongoing efforts of several academics to build deep learning algorithm-based stock price prediction systems were highlighted by Hong et al. [24]. To support informed decision-making, it is necessary to continuously monitor the highly volatile stock prices, which are influenced by a wide range of factors such as trading volume, news, revenue, and market dynamics. Because bidirectional Long Short-Term Memory (Bi LSTM) networks offer more accuracy than unidirectional LSTM networks, they were used to estimate market prices.

Upadhyay et al. [25] underscored the critical significance of stock markets within the international financial system, concentrating on their influence on both economic expansion and stability. They centered on the application of deep learning algorithms to improve the prediction of stock value. A comparative analysis was undertaken to evaluate the performance and precision of LSTM and Recurrent Neural Networks (RNN) algorithms in the context of stock price estimation. The objective of the research was to investigate the capacity of deep learning algorithms to establish a stock market environment that is more dependable and predictable. The utilization of historical market data obtained from the Alpha Vault API was employed to assess the efficacy of RNN and LSTM models in the prediction of stock prices. The results indicated that LSTM exhibited greater accuracy and was more appropriate for forecasting stock prices in comparison to RNN, which faced specific obstacles. In its entirety, the study enhanced comprehension regarding the utilization of deep learning algorithms in the analysis of the stock market, thereby enabling well-informed investment choices that aim to mitigate risks and optimize returns.

Intricate network issues were examined by Cao et al. [26] in stock market analysis and volatility prediction. Using multivariate stock time series data from the DJIA, S&P 500, and NASDAQ, pattern networks were built. Network topology features including strength, shortest path length, average degree centrality, and proximity centrality were shown to be useful in predicting changes in the market. Afterward, these topological characteristic variables were subjected to the K-nearest neighbors (KNN) and support vector machine (SVM) algorithms for stock volatility prediction. The best models for both algorithms were found utilizing search and cross-validation; SVM produced prediction accuracy rates higher than 70% for the assessed indices. According to their research, SVM algorithms beat KNN algorithms in the prediction of stock price volatility, indicating the potential benefits of machine learning and complex network analysis.

Srivinay et al. [27] recognized that the fluctuation of stock prices, which is affected by a multitude of elements including geopolitical tensions, corporate earnings, and commodity costs, presents traders with difficulties in precisely estimating volatility. To mitigate this difficulty and assist investors in reducing risk, they suggested the implementation of a hybrid stock prediction model that integrates the Prediction Rule Ensembles (PRE) method with a Deep Neural Network (DNN). Moving averages and other stock technical indicators were initially utilized to identify uptrends. Following this, prediction rules were generated using the PRE technique, and those resulting in the smallest RMSE were chosen. Following hyperparameter fine-tuning, a three-layer DNN was subsequently applied to stock prediction. The performance of the hybrid model was assessed using MAE and RMSE metrics. The results indicated that the hybrid model outperformed individual prediction models such as DNN and ANN, with a significant RMSE score improvement of 5% to 7%. They applied Indian stock price data to authenticate the suggested methodology.

As per the statement made by Jadhavrao et al. [28], they aimed to examine approaches to stock forecasting that utilized neural network techniques. When they first looked into the possibility of using neural networks to forecast stock market values, they emphasized how well they worked to find patterns in chaotic and nonlinear systems. Additionally, an examination was carried out to compare artificial intelligence algorithms with traditional and contemporary approaches used to forecast stock market trends. Lastly, by analyzing forecast criteria and factors influencing the Indian stock market, the algorithm's effectiveness was assessed on a variety of equities listed in both the US and India.

C. Research Gaps and Contributions

Despite extensive research on machine learning algorithms designed for stock price prediction, a direct comparison of the effectiveness and performance of these models does not appear to exist. Although some research studies have utilized machine learning algorithms, they may not have conducted comprehensive investigations into the potential benefits of integrating advanced optimization techniques to enhance the precision of forecasts. The literature review focuses primarily on specific algorithms or models, with limited exploration of the potential benefits associated with ensemble methods or hybrid models. Although these methodologies have the potential to generate more accurate forecasts by leveraging the merits of numerous algorithms, they are not investigated in the review. While the study explored various methodologies for predicting stock prices, there seems to be a scarcity of empirical research and validation of these models using datasets of historical stock prices. The main contributions of the study are as follows:

- The prediction accuracy of the proposed methodology, which integrates HGBoost with optimization techniques including PSO, SMA, and GWO, is significantly enhanced in comparison to conventional models. Through the utilization of these methods in concert, the model attains greater R^2 scores, which serve as an indicator of a more accurate prognosis regarding forthcoming stock price patterns.
- An additional noteworthy contribution of this research is the comparative evaluation of various optimization methodologies when utilized in conjunction with HGBoost. The results underscore GWO's superiority as an optimizer in maximizing prediction accuracy, thereby offering significant contributions to future research and practical implementations.
- The implications of the research findings extend to algorithmic trading strategies that seek to maximize investment decision efficiency. The proposed methodology enhances the precision of stock price forecasts, empowering investors and traders to make more knowledgeable and prompt decisions. As a result, portfolio performance is improved and risks are mitigated.

II. Materials and Methods

Predictions about the stock market give investors useful information that helps them make wise investing choices. Accurate projections are helpful for risk management and portfolio optimization for both institutional and individual investors. Precise forecasts enable investors to evaluate the risks attached to their investments. Investors can reduce risk and safeguard their cash by making decisions based on their awareness of possible market fluctuations. Therefore, developing a model for accurately predicting economic market movements is very important.

A. Histogram-based Gradient Boosting Regressor

One unique member of the Gradient Boosting Regressor family is the HGBoost, which uses histograms to speed up the computation of gradients and Hessians associated with the loss function [29], which is shown in Fig. 1. The process starts with fitting a regressor to the training dataset and then goes on to fit more regressors to the initial models' residual errors [17]. The combination of these ineffective learners is designed to create the final algorithm. This algorithm's primary goal is to reduce the loss function:

$$L = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{1}$$

During each iteration, the procedure involves fitting a weak learner, denoted as $h_{t(x)}$ to the residual errors derived from the preceding regressors. The dataset undergoes partitioning into bins, which is shown in detail in Fig. 1, guided by the decision tree of the weak learner and the values of the input features. Subsequently, the method leverages the histogram data to directly compute the gradients and Hessians of the loss function, as opposed to relying on approximations. The determination of the learner's weight is then conducted through precise calculations employing these gradients and Hessians. Notably, one notable advantage offered by histogram gradient boosting lies in its inherent ability to handle missing values and categorical attributes by intuitively creating new bins for each distinct category or absent data point. The final model is derived through a weighted averaging of each weak learner.

$$\hat{y}(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{2}$$

where α_t is the learner's weight for the t-th weak learner.

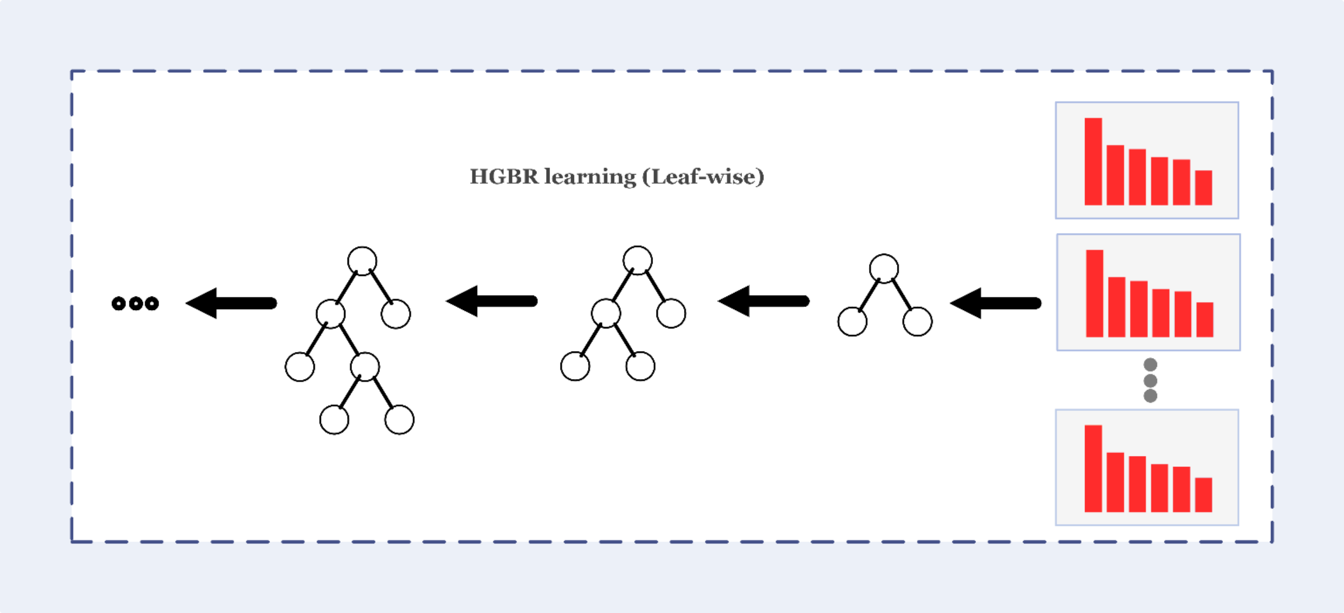

Fig. 1. Description of the Histogram-based gradient boosting regressor

B. Optimization Algorithms

The investigation continues to a critical point where each network's hyperparameters require careful adjustment. The foundation of this optimization project is the combination of three prominent and different models: PSO, SMA, and GWO.

C. Particle Swarm Optimization

In order to discover the best answers to optimization problems, people or particles in PSO, an algorithm inspired by nature, alter their locations in a multidimensional space, mimicking the social behavior of fish or birds in flocks. PSO can be useful for problems with complicated and nonlinear solution spaces and is frequently utilized for continuous optimization tasks [30].

The study of social behaviors seen in aquatic and avian species is the source of PSO. The effectiveness of this heuristic technique has been shown in examining continuous and multidimensional domains to find answers to optimization and search conundrums. The groundbreaking research conducted in the 1990s by James Kennedy and Russell Eberhart is credited with helping to conceptualize the PSO approach [30]. Every method placement in this algorithm is considered a possible solution inside a D-dimensional search space. The best-performing particle's location and the ideal position discovered have an impact on the particles, causing them to reposition themselves. Particles adjust their velocities using the following equation, which is utilized by the PSO algorithm:

$$v_{id}^{t+1} = v_{id}^{t} + C_1 r_1^t \left(Pbest_{id}^t - x_{id}^t\right) + C_2 r_2^t \left(Gbest_{id}^t - x_{id}^t\right) \tag{3}$$

where v_{id}^{k} is the ith particle's speed during a specific time iteration in a d-dimensional search space. In $Pbest_{id}^t$ and $Gbest_{id}^t$, respectively, the ideal particle and location for the ith individual and iteration t are shown. While C_1 and C_2 are parameters used to adjust particle speed, r_1^t and r_2^t are random values between 0 and 1. Furthermore, the particles in the PSO algorithm adjust their locations by using the following equation:

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \tag{4}$$

In this instance, x_{id}^t represents the ith particle's location in iteration t and in a d-dimensional search space.

D. Slime Mold Algorithm

In the year 2020, Li et al. [19] introduced the SMA, a computational model primarily designed to emulate the behavioral and morphological transformations observed in Physarum polycephalum during its foraging activities. The SMA incorporates the concept of using weights to simulate both positive and negative feedback mechanisms that occur during the slime mold's foraging process, ultimately leading to the emergence of three distinct morphological forms within the slime mold. Physarum polycephalum, a eukaryotic organism, thrives in cold and humid environments, with its primary source of sustenance being Plasmodium. During its active feeding phase, the slime mold's organic matter seeks out food sources, envelops them, and releases enzymes to facilitate their decomposition. To support the flow of cytoplasm, the leading edge of the migrating cell moves in specific sectors, while the trailing end forms a network of interconnected veins. The slime mold can construct such venous networks based on the characteristics of various food sources it encounters.

The mathematical formula employed to describe the behavior of the slime mold forms the fundamental basis of the SMA approach, which can be applied across a wide range of fields and domains.

$$\overrightarrow{X(t+1)} = \begin{cases} \overrightarrow{X_b(t)} + \overrightarrow{v_b} \,.\left(\overrightarrow{W} \,.\overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}\right) & r < p \\ \overrightarrow{v_c}.\overrightarrow{X(t)} & r \geq p \end{cases} \quad (5)$$

Whereas $X(t)$ and $X(t+1)$ are the locations of the slime mold in repetitions t and $t+1$, respectively, and $X_b(t)$ represents the area of the slime mold with the highest concentration of odor at this specific instant. $X_A(t)$ and X_B display two randomly chosen spots for slime mold and v_b is a variable that changes over time [- a , a]($a = \operatorname{arctanh}(-(\frac{t}{\max\_t}) + 1)$), If v_c is a decreasing linear the definition of p is as follows: if v_c is a parameter that decreases linearly from 0 to 1, and r is a random number between 0 and 1:

$$p = tanh|S(i) - DF| \quad i = 1, 2, \dots , n \quad (6)$$

$S(i)$ denotes the fitness of $\overrightarrow{X}$ and DF denotes the iteration that is overall the fittest. The following is a description of the weight W equation:

$$\overrightarrow{W(smell\ \imath ndex(l))} = \begin{cases} 1 + r.\log\left(\frac{bF-S(i)}{bF-wF} + 1\right), condition \\ 1 - r.\log\left(\frac{bF-S(i)}{bF-wF} + 1\right)\ , others \end{cases} \quad (7)$$

$$smell\ index = sort(S) \quad (8)$$

$S(i)$ denotes the first half of the population, bF denotes the best fitness, wF denotes the worst fitness, and the smell index represents the values of the sorted fitness. The position of the slime mold may be altered using the following equation:

$$\overrightarrow{X^*} = \begin{cases} rand(UB - LB) + LB & rand < z \\ \overrightarrow{X_b(t)} + \overrightarrow{v_b}.\left(\overrightarrow{W} \,.\overrightarrow{X_A(t)} - \overrightarrow{X_B(t)}\right) & r < p \\ \overrightarrow{v_c}.\overrightarrow{X(t)} & r \geq p \end{cases} \quad (9)$$

where LB and UB are the lower and upper limits of the finding interval, respectively, and z is an integer between 0 and 0.1.

E. Grey Wolf Optimization

The Gray Wolf Optimizer is a unique optimization approach that has been developed using a meta-heuristic technique. The methodology, which emulates the societal organization and hunting strategies used by gray wolves, was first introduced by Mirjalili et al. Its overall structure is illustrated in Fig. 2., which starts with the placement of the search agents in the problem space. Then after evaluating the fitness value of each agent, the alpha, beta, and delta are selected. If the maximum iteration is reached, the best values will be chosen. [20]. Alpha is considered the optimal alternative, whilst Omega represents the last contender within the leadership hierarchy. This hierarchy has four possibilities, namely Alpha, Beta, Delta, and Omega, whose position has been determined in Fig. 3 based on their position and distance to the prey where the nearest wolf is alpha, the second one is beta, and the remained wolves are delta.

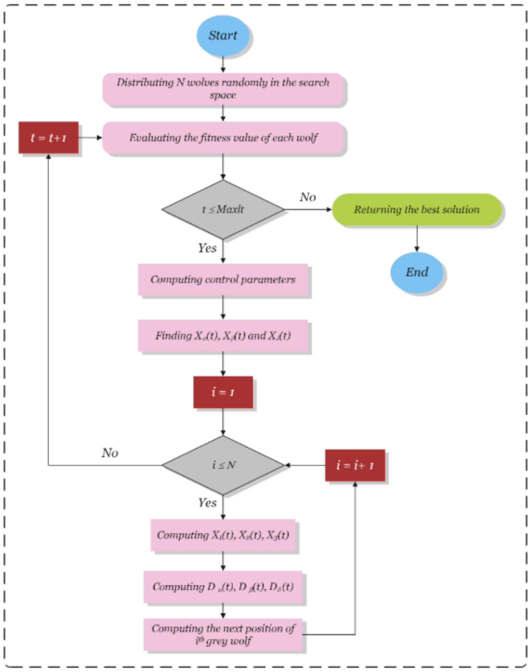
Start
Distributing N wolves randomly in the search space
t = t+1
Evaluating the fitness value of each wolf
t ≤ MaxIt
No
Returning the best solution
End
Yes
Computing control parameters
Finding $X_\alpha(t)$, $X_\beta(t)$ and $X_\delta(t)$
i = 1
No
i ≤ N
i = i+ 1
Yes
Computing $X_1(t)$, $X_2(t)$, $X_3(t)$
Computing $D_\alpha(t)$, $D_\beta(t)$, $D_\delta(t)$
Computing the next position of i^{th} grey wolf

Fig. 2. Grey Wolf Optimization Flowchart

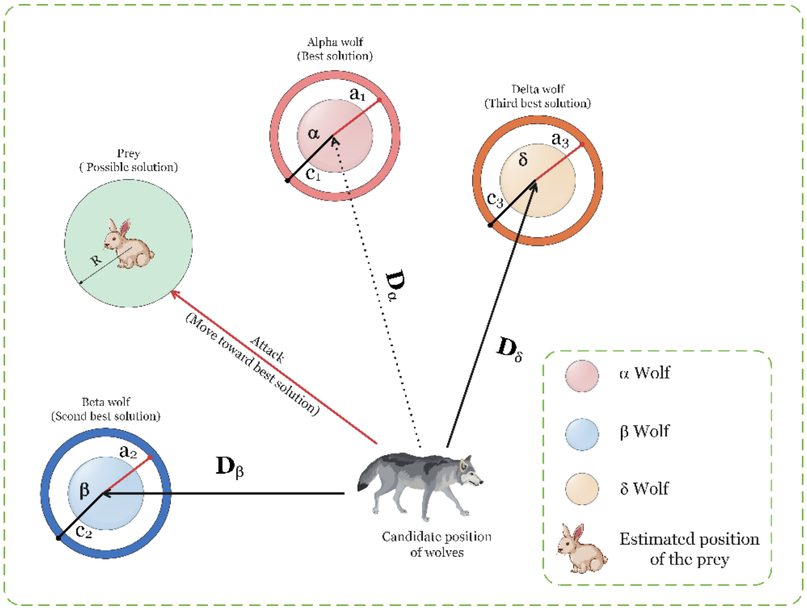

Fig. 3. Position of the wolves in nature

The approach utilizes three main hunting strategies to imitate the behavior of wolves: prey pursuit, prey enclosure, and prey assault. To simulate the hunting behavior of gray wolves in their natural habitat, the following link was employed:

$$\vec{D} = \left|\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)\right|$$

$$\vec{X} \mid (t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D} \qquad (10)$$

in which, $\vec{X}_p$ denotes prey location, $\vec{D}$ denotes movement, $\vec{A}$ and $\vec{C}$ denotes coefficient vectors, t is the current iteration, and $\vec{X}$ denotes the position of a gray wolf. The following relationships are used to construct the coefficient vectors ($\vec{A}$ and $\vec{C}$):

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \qquad (11)$$

The spatial allocation of novel search representatives pertaining to omegas is modified by using data derived from alpha, beta, and delta in the following manner:

$$\vec{D}_a = \left|\vec{C}_1 \cdot \vec{X}_a - \vec{X}\right|, \vec{D}_\beta = \left|\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}\right|, \vec{D}_\delta = \left|\vec{C}_3 \cdot \vec{X}_s - \vec{X}\right| (12)$$

$$\vec{X}_1 = \vec{X}_a - \vec{A}_1 \cdot \vec{D}_u, \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta, \vec{X}_3 = \vec{X}_b - \vec{A}_3 \cdot \vec{D}_\delta (13)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} (14)$$

where the subscripts $\alpha, \beta,$ and δ represent the wolves, who must launch a last assault to finish the mission. An is a random variable that lies between -2a ⃗ and 2a ⃗, whereas a ⃗ is utilized to simulate the previous assault by altering a value from 2 to 0. Therefore, lowering $\vec{a}$ would likewise result in lowering $\vec{A}$. The wolves were coerced into clinging to their prey by $|\vec{A}| < 1$. Gray wolves hunt in packs and follow the leader wolf, splitting out to gather food and then coming together to attack. Wolves may separate in search of prey when $|\vec{A}|$ has a random value greater than unity. The GWO method relies heavily on two key configuration parameters, namely the wolf count and generation number. These parameters play a critical role in determining the algorithm's performance and effectiveness. The population of wolves accurately depicts the number of function evaluations through time, with each generation signifying the decisive actions of individual wolves. The total number of objective function evaluations will thus be equal to the product of the wolf population and the generation size.

$$OFEs = N_W \times N_G \qquad (15)$$

F. *Proposed Framewrok*

Fig. 4 represents the overall stages of the framework. Firstly, the daily datasets of the Alphabet were collected then these data underwent a thorough data preparation where they became normalized and split into train and test sets. Next, these data were fed to the HGBoost model, which wasn't optimized. Subsequently, three different optimizers were used to optimize the hyperparameters of the HGBoost model and it was found to be that the GWO-HGBoost model outperformed other models by obtaining the best values.

G. *Description of Dataset*

The goal of the dataset utilized in this study is to enable forecasting of Alphabet Inc. share prices over an extended period, spanning from 2015 to mid-2023. Accurate stock price forecasting is essential for financiers, investors, and decision-makers in the industry. This dataset contains the historical stock price data and related characteristics needed to carry out prediction analyses. Stock exchanges and financial news sites are the main sources of financial market data in the collection. The historical daily stock share values of Alphabet Inc. for the given period were collected. The parameters used in this paper's dataset are several bits of data about Alphabet Inc. shares that are accessible on each day of trading between 2015 and mid-2023.

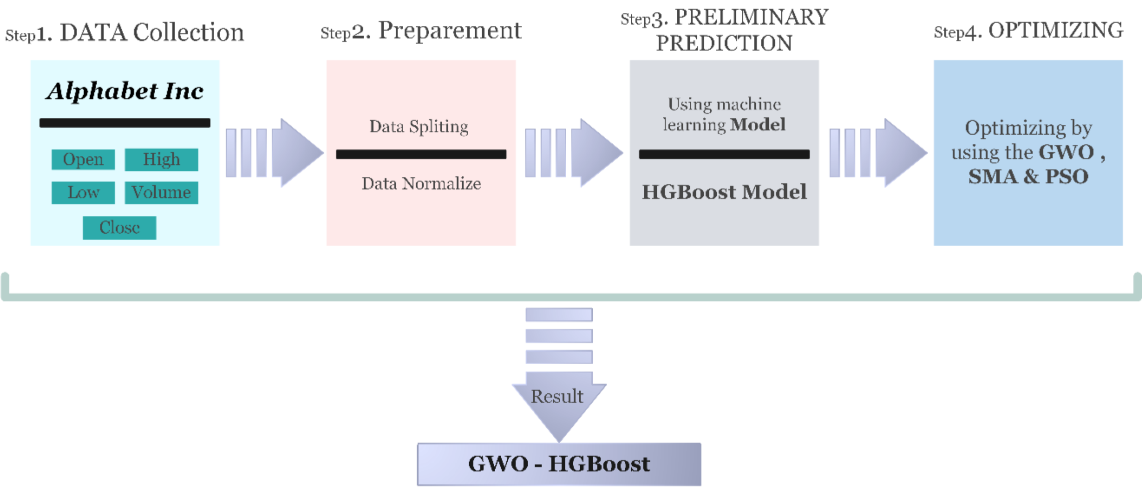

Fig. 4. Overall stages of the suggested framework

This encompasses various data points, such as the date, the opening price when the trading day begins, the closing price when the trading day ends, the highest share price reached during the day, the lowest share price during the day, and the trading volume, which signifies the total number of shares traded in a day. Stringent data preprocessing steps were employed to ensure data quality and consistency before undertaking any predictive analyses. Additionally, data normalization was conducted to facilitate precise modeling and forecasting. Data normalization involves scaling numerical variables to a standardized range, typically between 0 and 1, or with a mean of 0 and a standard deviation of 1. This ensures that variables with varying units or magnitudes are treated uniformly in analytical or modeling tasks. The size of input variables has an impact on the performance of many machine learning techniques, and normalizing the data can enhance the performance and convergence of these algorithms.

The reason for using the data normalization technique is that when working with normalized data, several machine learning optimization techniques converge more quickly. This can minimize the number of computational resources needed and expedite the training process. The normalization formula is expressed in the following equation:

$$\text{XScaled} = \frac{(X-Xmin)}{(Xmax-Xmin)} \qquad (16)$$

Data splitting is a common procedure used to evaluate a machine learning model's capacity to handle fresh, untested data. By training the model on one dataset segment and testing it on another, this approach enables to assess the model's performance in real-world scenarios. By separating, it is easy to ascertain if the model has truly learned from the data and identified patterns by dividing it into training and testing subsets or if it only depends on information from its training data. Fig. 5 displays a detailed view of the complete training and testing dataset, where the training sets span from 2015 to approximately 2021, and the testing sets cover the period between 2021 to 2023.

H. Statistical Analysis of the Data

The statistical outcomes from the acquired data are presented in Table I. When characterizing the attributes of a dataset, descriptive statistics such as the count, average, median, skewness, standard deviation, kurtosis, variance, maximum, and minimum values are employed.

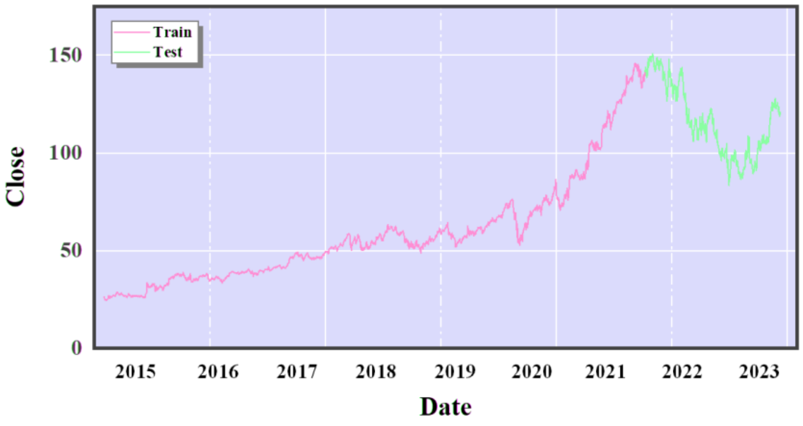

Fig. 5. The complete depiction of the dataset while in the training and testing phases

TABLE I. STATISTICAL RESULT OF THE PRESENTED DATASET

	Open	High	Low	Volume	Close
count	2137	2137	2137	2137	2137
Mean	70.05219	70.81457	69.3428	32.59751	70.09629
Std.	34.54605	34.97686	34.14654	15.6062	34.55914
Min	24.66478	24.7309	24.31125	6.936	24.56007
50%	58.4235	58.9	57.871	28.734	58.4095
Max	151.8635	152.1	149.8875	223.298	150.709
Skew	0.746243	0.736992	0.747426	2.879365	0.741179
kurtosis	-0.6277	-0.65576	-0.62251	16.58048	-0.64157
variance	1193.43	1223.381	1165.986	243.5536	1194.334

Through mathematical techniques for summarizing data, several fundamental statistics are calculated. The mean, also known as the average, is ascertained by summing all the values within a dataset and dividing the sum by the total count of values. The median, in turn, is determined by arranging the dataset in ascending order and identifying the middle value. In cases where the dataset comprises an even number of values, the median is computed as the average of the two middle values. Notably, the median is less susceptible to the influence of outliers or extreme values compared to the mean. The skewness of a dataset is a measure that characterizes the asymmetry of its distribution. This statistical metric provides insight into whether the data exhibits symmetry, a positive skew to the right, or a negative skew to the left. Specifically, a skewness value of 0 signifies a perfectly symmetric distribution. To assess the dispersion of data points around the mean, the standard deviation is employed. This metric quantifies how much individual data points deviate from the mean. A higher standard deviation indicates greater variability within the dataset. Mathematically, the standard deviation is represented as the square root of the variance. The maximum value within a dataset corresponds to the highest value present among all data points. Conversely, the minimum value represents the lowest value within the dataset. These fundamental statistics are vital for comprehensively characterizing and summarizing the features of a dataset in quantitative terms.

I. Assessment Criteria

In the evaluation of models, algorithms, and data-driven solutions in diverse domains such as machine learning, data science, and business analytics, the utilization of evaluation metrics is paramount. These metrics serve as essential instruments for quantitatively assessing the performance and effectiveness of a model or approach in achieving its intended objectives. This research employs specific criteria, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), and R-squared (R²). MAE quantifies the average absolute difference between predicted and actual values, providing a straightforward means of gauging prediction accuracy. RMSE, the square root of MSE, furnishes a comprehensible measure expressed in the same units as the target variable, enhancing interpretability. R-squared, denoted as R², elucidates the extent to which the model accounts for the variability in the target variable. It ranges from 0 to 1. These metrics are fundamental for the rigorous assessment and quantification of the performance of models and data-driven solutions, ensuring objective and robust evaluations in diverse fields.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{17}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{18}$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{19}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{20}$$

III. RESULT AND DISCUSSION

To evaluate each prediction model's accuracy in predicting the variable of interest, this paper used three different models in the study: HGBoost, PSO- HGBoost, SMA- HGBoost, and GWO- HGBoost. The predicted efficacy of the models was assessed using a range of performance criteria. According to this investigation, GWO- HGBoost consistently performed better in terms of reliability and accuracy of predictions than the other models.

The performance metrics for each model, including R², RMSE, and MAE, are summarized in Table II. Predictive model precision and goodness of fit are frequently evaluated using these criteria.

As can be seen in Table II, GWO- HGBoost showed the lowest RMSE and MAE values, indicating that its predictions were more accurate than those of the other models. Additionally, it had the greatest R^2 value, demonstrating the greater predictive power of GWO- HGBoost by explaining a greater percentage of the variance in the target variable. GWO enhances predictive performance through the optimization of model fitting. By performing this optimization, GWO-HGBoost could potentially enhance its ability to optimize model parameters to generate more accurate predictions. The performance of GWO-HGBoost indicates that it is more adaptable to market changes than alternative methods. GWO-HGBoost enhances its prognostic capabilities through the incorporation of stock price dynamics and resistance to market trends. GWO-HGBoost demonstrates strong performance across datasets, as evidenced by its exceptional accuracy on both the training and test sets. This implies that GWO-HGBoost exhibits dependability and efficacy in practical contexts due to its capacity to retain its predictive capability and effectively extrapolate to unobserved data. In conclusion, in terms of predictive accuracy, precision, adaptability, and robustness, GWO-HGBoost surpasses alternative methods. Using optimization and sophisticated modeling, GWO-HGBoost more accurately predicts stock market trends, making it a potentially effective method for financial decision-making and risk management.

As shown in Table II, among the optimization methods, SMA has better results than PSO, and GWO has much better results than SMA, which has made it the best optimal method for optimizing the hyperparameters of the HGBoost. The fit comparison between the real data points and the forecasts produced by the four models, HGBoost, PSO- HGBoost, SMA-HGBoost, and GWO- HGBoost, is presented in Fig. 6 during Train and in Fig. 7 during Testing. Every data point in the collection is an observation, and the lines or curves show the expected values produced by the corresponding models.

When Fig. 6 and Fig. 7 are closely examined, it is observed that GWO- HGBoost consistently shows the best alignment with the real data points; that is, the red data are most closely resembled by it even in the reversal points of the market, it can be seen that the proposed method resembles the actual curve and this indicates and proves the efficiency of the GWO-HGBoost. This is consistent with the numerical performance indicators previously displayed in Table II, where GWO- HGBoost was found to have produced the lowest MAE, RMSE, and the greatest R^2 of all the models. The results of Table II are also shown in Fig. 8 and Fig. 9, in which the obtained values during train and testing for four different metrics using four different algorithms are provided.

TABLE II. PERFORMANCE METRICS FOR PREDICTION MODELS

MODEL/Metrics	TRAIN SET				TEST SET			
	R^2	*RMSE*	*MAPE*	*MAE*	R^2	*RMSE*	*MAPE*	*MAE*
HGBoost	0.971	4.634	3.647	2.793	0.964	3.475	2.722	3.199
PSO-HGBoost	0.983	3.481	2.937	2.151	0.973	3.005	2.048	2.331
SMA-HGBoost	0.987	3.067	4.393	2.379	0.981	2.524	1.728	2.035
GWO-HGBoost	0.991	2.515	2.721	1.997	0.988	2.001	1.305	1.542

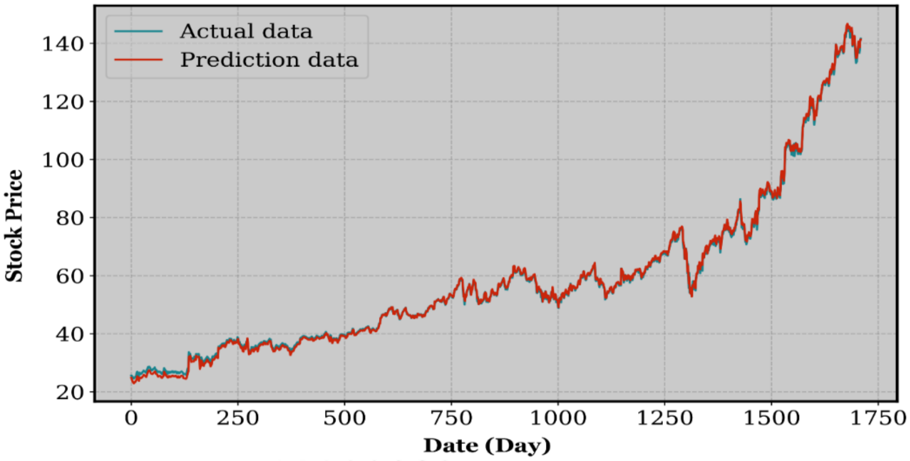

Fig. 6. The comparison between the actual data and the predictions made by GWO- HGBoost during training

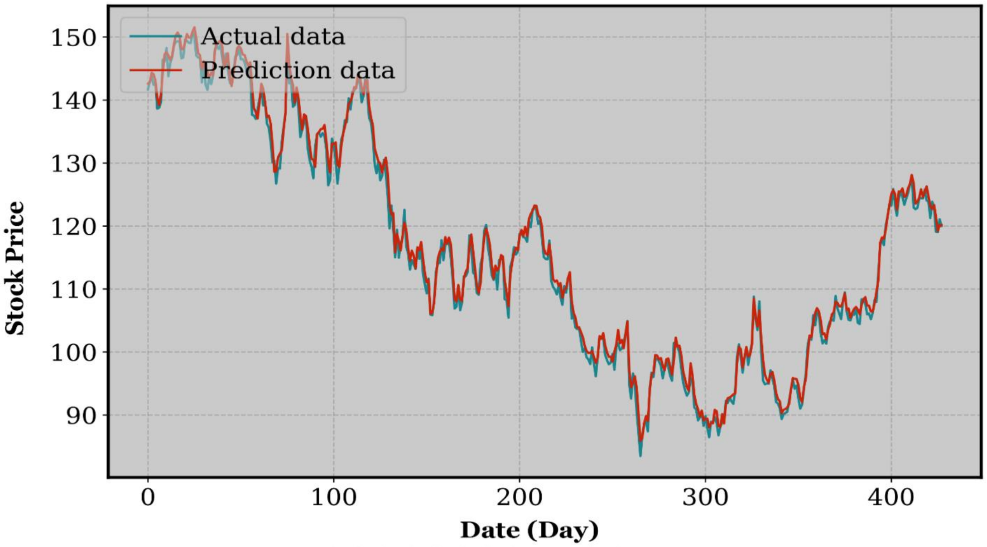

Fig. 7. The comparison between the actual data and the predictions made by GWO- HGBoost during the Test

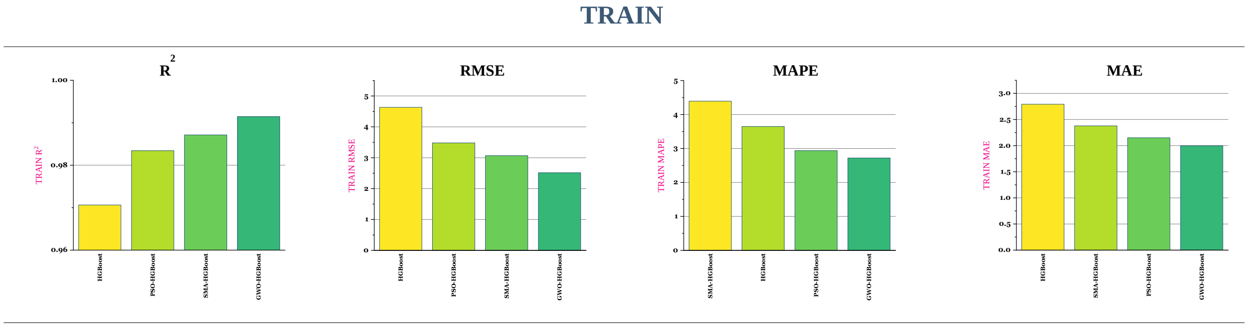

Fig. 8. The results of the optimized model by PSO, SMA, and GWO and the description of their performance during training

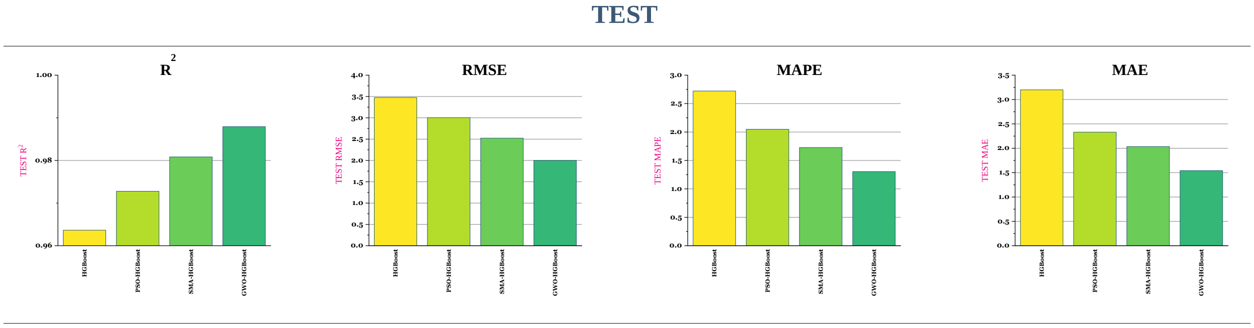

Fig. 9. The results of the optimized model by PSO, SMA, and GWO and the description of their performance during Testing

Validation procedures and comparisons with previously published relevant literature are critical components in assessing the reliability and importance of a research inquiry. Furthermore, they collaborate to situate the research within a broader context, thereby ensuring the reliability and precision of the study's results. The current assessment examines, as demonstrated in Table III, the prognostic capacities of various models concerning the behavior of the stock market. Out of the models that were assessed, the GWO- HGBoost model emerges as the most effective with a coefficient of determination of 0.988. This value surpasses that of every other method included in the list, which comprises Linear Regression, SVM, different iterations of LSTM, DNN, and combinations of DNN and LSTM. The remarkable degree of precision observed in the forecasts of stock market trends underscores the efficacy and dependability of the GWO- HGBoost model in capturing the intricacies intrinsic to fluctuations in stock prices. Through the integration of Grey Wolf Optimization and histogram-based Gradient Boosting, the GWO- HGBoost model achieves enhanced predictive performance by optimizing model fitting and hyperparameters. By accelerating the computation of gradients and Hessians associated with the loss function, histograms contribute to the improvement of the model's efficiency and precision. In addition, the GWO- HGBoost model's adaptability and resilience in various market conditions are enhanced by its ensemble learning methodology and capability to handle missing values and categorical attributes. In conclusion, the GWO- HGBoost model demonstrates its efficacy and consistency in forecasting the stock market, as indicated by its performance in Table III. The potential significance of this method in financial decision-making and risk management is highlighted by its exceptional accuracy, which provides analysts and investors with invaluable insights.

TABLE III. A MODEL EVALUATION OF PREVIOUS STUDIES IS PROVIDED

Authors	Methods	R^2
Abdul et al. [31]	Linear regression	0.735
	SVM	0.931
	MLS-LSTM	0.950
Zhu et al. [32]	LSTM	0.689
	EMD-LSTM	0.870
	CEEMDAN-LSTM	0.903
	SC-LSTM	0.687
	EMD-SC-LSTM	0.911
	CEEMDAN-SC-LSTM	0.920
Nayak et al. [33]	DNN and LSTM	0.972
Jin et al. [34]	LSTM	0.981
Current study	GWO-HGBoost	0.988

IV. CONCLUSION

In conclusion, by utilizing enormous volumes of data and potent algorithms to produce more accurate forecasts, machine learning has completely transformed the field of stock prediction. Machine learning algorithms can recognize intricate patterns and adjust to shifting market conditions, which can greatly improve investment methods. But it's important to understand that stock prediction is still a difficult and unpredictable task because a lot of things, like human behavior and unanticipated events, affect financial markets. Combining machine learning with solid financial knowledge and a deep comprehension of market dynamics is essential to maximizing its potential. The future of stock prediction is likely to be defined by the collaboration between humans and machines as the field develops. Not to mention, the creation and evaluation of the prediction model illustrated how important it is to use data-driven insights to make trustworthy decisions. This illustrates the possible applications of predictive analytics in a variety of industries as well as the benefits of a data-centric approach in the contemporary, quickly changing corporate environment. To enable traders and investors to use these algorithms to make purchases on the appropriate day and at the appropriate price, the goal of this study was to develop models that could more accurately predict stock prices.

This paper's conclusions included the following:

- The order in which the normalization and data preparation were finished could influence the presentation of the prediction model. After that, the data was prepared for the next phases in the selected model's analysis.
- Selecting the best model, evaluating the outcomes, and then modifying the model's hyperparameters to increase the supplied model's efficiency.
- By comparing the output of multiple optimizers, the most accurate optimization has been determined to be the main optimizer of the model. The GWO- HGBoost approach yields the best results when compared to PSO- HGBoost and SMA- HGBoost. The results are 0.973, 0.981, and 0.988 for PSO- HGBoost, SMA- HGBoost, and GWO-HGBoost by use of R^2 evaluation criteria.

The efficacy of predictive models is significantly contingent upon the accessibility and caliber of historical data. A lack of sufficient or dependable data sources can impede the precise depiction of market dynamics. Although machine learning algorithms have the potential to enhance the accuracy of predictions, their intricate nature frequently presents difficulties in interpretation, particularly for professionals in the financial industry. The task of adjusting models to diverse market conditions continues to present difficulties, and additional investigation is required to optimize the integration of optimization methods such as PSO, SMA, and GWO with HGBoost. Particularly with complex algorithms and limited datasets, there is a risk of overfitting; therefore, exhaustive cross-validation and testing on out-of-sample data are required for an accurate evaluation.

By incorporating supplementary data sources including news articles, social media sentiment, and macroeconomic indicators, the predictive capabilities of the models could be significantly improved, resulting in a more holistic comprehension of market dynamics. It is of the utmost importance to devise techniques that improve the interpretability of machine learning models while maintaining their predictive accuracy. Methods such as feature importance analysis and model explanation frameworks have the potential to offer significant insights regarding the determinants that influence model predictions. The development of adaptive algorithms capable of dynamically modifying model parameters to account for evolving market conditions has the potential to enhance the resilience and dependability of the models within real-time trading environments. Investigating ensemble learning methodologies that integrate numerous models, such as conventional statistical techniques and machine learning algorithms, may yield additional benefits in terms of enhanced prediction precision and reduced model biases.

REFERENCES

[1] Y. Baek and H. Y. Kim, “ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module,” *Expert Syst Appl*, vol. 113, pp. 457–480, 2018, doi: https://doi.org/10.1016/j.eswa.2018.07.019.

[2] K. Pardeshi, S. S. Gill, and A. M. Abdelmoniem, “Stock Market Price Prediction: A Hybrid LSTM and Sequential Self-Attention based Approach,” 2023. doi: 10.48550/arxiv.2308.04419.

[3] L. N. Mintarya, J. N. M. Halim, C. Angie, S. Achmad, and A. Kurniawan, “Machine learning approaches in stock market prediction: A systematic literature review,” *Procedia Comput Sci*, vol. 216, pp. 96–102, 2023, doi: 10.1016/j.procs.2022.12.115.

[4] Y. Xu, J. Liu, F. Ma, and J. Chu, “Liquidity and realized volatility prediction in Chinese stock market: A time-varying transitional dynamic perspective,” *International Review of Economics and Finance*, vol. 89, no. PA, pp. 543–560, 2024, doi: 10.1016/j.iref.2023.07.083.

[5] S. Mukherjee, B. Sadhukhan, N. Sarkar, D. Roy, and S. De, “Stock market prediction using deep learning algorithms,” *CAAI Trans Intell Technol*, vol. 8, no. 1, pp. 82–94, 2023, doi: 10.1049/cit2.12059.

[6] D. Shah, H. Isah, and F. Zulkernine, “Stock market analysis: A review and taxonomy of prediction techniques,” *International Journal of Financial Studies*, vol. 7, no. 2, 2019, doi: 10.3390/ijfs7020026.

[7] W. H. Bangyal, M. Iqbal, A. Bashir, and G. Ubakanma, “Polarity Classification of Twitter Data Using Machine Learning Approach,” in *2023 International Conference on Human-Centered Cognitive Systems (HCCS)*, IEEE, 2023, pp. 1–6.

[8] E. S. Olivas, J. D. M. Guerrero, M. Martinez-Sober, J. R. Magdalena-Benedito, and L. Serrano, *Handbook of research on machine learning applications and trends: Algorithms, methods, and techniques: Algorithms, methods, and techniques*. IGI global, 2009.

[9] S. B. Kotsiantis, “Decision trees: a recent overview,” *Artif Intell Rev*, vol. 39, pp. 261–283, 2013.

[10] L. Breiman, “Random forests,” *Mach Learn*, vol. 45, pp. 5–32, 2001.

[11] P. Chhajer, M. Shah, and A. Kshirsagar, “The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction,” *Decision Analytics Journal*, vol. 2, no. November 2021, p. 100015, 2022, doi: 10.1016/j.dajour.2021.100015.

[12] A. Natekin and A. Knoll, “Gradient boosting machines, a tutorial,” *Front Neurorobot*, vol. 7, p. 21, 2013.

[13] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[14] W. H. Bangyal, S. Amina, R. Shakir, G. Ubakanma, and M. Iqbal, “Using Deep Learning Models for COVID-19 Related Sentiment Analysis on Twitter Data,” in *2023 International Conference on Human-Centered Cognitive Systems (HCCS)*, IEEE, 2023, pp. 1–6.

[15] S. Sun, Z. Cao, H. Zhu, and J. Zhao, “A survey of optimization methods from a machine learning perspective,” *IEEE Trans Cybern*, vol. 50, no. 8, pp. 3668–3681, 2019.

[16] B. Bischl *et al.*, “Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges,” *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 13, no. 2, p. e1484, 2023.

[17] A. Guryanov, “Histogram-based algorithm for building gradient boosting ensembles of piecewise linear decision trees,” in *Analysis of Images, Social Networks and Texts: 8th International Conference, AIST 2019, Kazan, Russia, July 17–19, 2019, Revised Selected Papers 8*, Springer, 2019, pp. 39–50.

[18] S. Pervaiz, Z. Ul-Qayyum, W. H. Bangyal, L. Gao, and J. Ahmad, “A systematic literature review on particle swarm optimization techniques for medical diseases detection,” *Comput Math Methods Med*, vol. 2021, 2021.

[19] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, “Slime mould algorithm: A new method for stochastic optimization,” *Future Generation Computer Systems*, vol. 111, pp. 300–323, 2020, doi: https://doi.org/10.1016/j.future.2020.03.055.

[20] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014, doi: https://doi.org/10.1016/j.advengsoft.2013.12.007.

[21] L. Y. Jun *et al.*, "Modeling and optimization by particle swarm embedded neural network for adsorption of methylene blue by jicama peroxidase immobilized on buckypaper/polyvinyl alcohol membrane," *Environ Res*, vol. 183, p. 109158, 2020.

[22] D. G. Bhalke, D. Bhingarde, S. Deshmukh, and D. Dhere, "Stock Price Prediction Using Long Short Term Memory," *SAMRIDDHI - A JOURNAL OF PHYSICAL SCIENCES, ENGINEERING & TECHNOLOGY; Vol 14 No Spl-2 issu (2022): A Journal of Physical Sciences, Engineering and Technology (2022);; 271-273 ; 2454-5767 ; 2229-7111*, Apr. 2022, [Online]. Available: https://myresearchjournals.com/index.php/SAMRIDDHI/article/view/11072

[23] Z. Su and B. Yi, "Research on HMM-Based Efficient Stock Price Prediction," *Mobile Information Systems, Vol 2022 (2022)*, Apr. 2022, doi: 10.1155/2022/8124149.

[24] S. Hong and J. Han, "Stock Price Prediction by Using BLSTM (Bidirectional Long Short Term Memory)," *Journal of Computational and Theoretical Nanoscience ; volume 18, issue 5, page 1614-1617 ; ISSN 1546-1955*, 2021, doi: 10.1166/jctn.2021.9603.

[25] N. K. Upadhyay, V. Singh, S. Singh, and P. Khanna, "Enhancing Stock Market Predictability: A Comparative Analysis of RNN And LSTM Models for Retail Investors," *Journal of Management and Service Science (JMSS); Vol. 3 No. 1 (2023); 1-9 ; 2583-1798*, Apr. 2023, [Online]. Available: https://jmss.a2zjournals.com/index.php/mss/article/view/42

[26] H. Cao, T. Lin, Y. Li, and H. Zhang, "Stock Price Pattern Prediction Based on Complex Network and Machine Learning," 2019, doi: 10.1155/2019/4132485.

[27] Srivinay, B. C. Manujakshi, M. G. Kabadi, and N. Naik, "A Hybrid Stock Price Prediction Model Based on PRE and Deep Neural Network," *Data, Vol 7, Iss 51, p 51 (2022)*, Apr. 2022, doi: 10.3390/data7050051.

[28] R. Jadhavrao, R. Sengupta, A. Patil, and R. S. Yadav, "Stock Market Trend Prediction using Artificial Intelligence Algorithm (RNN-LSTM) - Comparison with Current Techniques and Research on its Effectiveness in Forecasting in Indian Market and US Market," 2023, doi: 10.5281/zenodo.10864037.

[29] S. Md. M. Hossain and K. Deb, "Plant Leaf Disease Recognition Using Histogram Based Gradient Boosting Classifier," in *Intelligent Computing and Optimization*, P. Vasant, I. Zelinka, and G.-W. Weber, Eds., Cham: Springer International Publishing, 2021, pp. 530–545.

[30] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, IEEE, 1995, pp. 1942–1948.

[31] A. Q. Md *et al.*, "Novel optimization approach for stock price forecasting using multi-layered sequential LSTM," *Appl Soft Comput*, vol. 134, p. 109830, 2023, doi: https://doi.org/10.1016/j.asoc.2022.109830.

[32] R. Zhu, G.-Y. Zhong, and J.-C. Li, "Forecasting price in a new hybrid neural network model with machine learning," *Expert Syst Appl*, vol. 249, p. 123697, 2024, doi: https://doi.org/10.1016/j.eswa.2024.123697.

[33] A. C. Nayak and A. Sharma, *PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26–30, 2019, Proceedings, Part II*, vol. 11671. Springer Nature, 2019.

[34] Z. Jin, Y. Yang, and Y. Liu, "Stock closing price prediction based on sentiment analysis and LSTM," *Neural Comput Appl*, vol. 32, pp. 9713–9729, 2020.