International Journal of Pattern Recognition and Artificial Intelligence

Enhancing gcForest with R-measure and three-objective-pruning for option price prediction

Rui Zhang
School of Management, Beijing Institute of Technology, Beijing, 100190, China.
1120220789@bit.edu.cn

Jiayi Li
School of Management, Beijing Institute of Technology, Beijing, 100190, China.
1120211776@bit.edu.cn

Qier Mu
School of Computer Science & Technology, Beijing Institute of Technology, Beijing, 100190, China.
1120220659@bit.edu.cn

Chenxue Yang
Agricultural Information Institute, CAAS, Beijing, 100190, China.
yangchenxue@caas.cn

The volume of option data is large and growing rapidly, characterized by a low signal-to-noise ratio, high dimensionality, nonlinearity, and computational complexity, which complicates accurate predictions of option price trends. In this paper, we collect historical data of Standard & Poor's 500 Index Options from 1998 to 2021, including 11 option characteristics and 7 macroeconomic features. The Option-Factor dataset (Optfa) is initially processed through normalization, missing value imputation, and extreme value replacement. We apply the multi-grained cascade forest algorithm (gcForest) to predict the option prices of the Optfa dataset. To extract meaningful features from the option data using prior knowledge, we construct the feature-enhanced R-Optfa dataset by combining the R-factor from a theoretical pricing model with the gcForest algorithm. We then propose the enhanced ADgcForest algorithm, which incorporates clustering, dimensionality reduction, and three-objective optimization (TOOCEP) pruning operations. The ADgcForest algorithm achieves an accuracy of 99.78%, with a test time of 4.3 seconds. Compared to gcForest, ADgcForest improves accuracy by 4.42% and reduces test time by approximately 0.15 seconds. Given the time-sequential nature of option data, we select time series models such as RNN, LSTM, transformer, and the VQ-VAE coding model to predict the option prices of the R-Optfa dataset. Their respective accuracies are 65.67%, 78.47%, 79.89%, and 85.34%. The proposed ADgcForest-R algorithm demonstrates high accuracy and strong generalization ability in forecasting option prices.

Keywords: Option Price Prediction, gcForest, R-measure, time series model, Tri-objective Optimization (TOOCEP)

1. Introduction

In the contemporary financial sector, predicting option prices has become a key research area. In 2023, the global trading volume of options reached 137.3 billion contracts, representing a 64% increase compared to the previous year. The sheer volume of option data is immense and expanding rapidly, contributing to the growing significance of the options market. As a result, the challenges of option pricing and risk evaluation are becoming increasingly complex. Traditional pricing models, such as the Black-Scholes Model for European Options and the Binomial Lattice Algorithm for American Options, struggle to accurately reflect real-world market dynamics and to support flexible investment choices under uncertain conditions.

With the rise of machine learning, numerous approaches have been effectively applied to address the challenge of large-scale option price prediction, thanks to their capacity to handle vast datasets and deliver precise analyses. For instance, [23] introduces a prediction model for Bitcoin and gold prices that successfully forecasts option prices by combining time series analysis with machine learning techniques. To address the limitations of conventional option pricing models, [24] integrates the option data prediction framework with BP neural networks and Support Vector Machines (SVM). Consequently, the fusion of deep learning with alternative mathematical techniques offers a more efficient and realistic approach to option price prediction.

As is widely recognized, option data spans a long period, with price trends that evolve over time. Option price series often exhibit self-correlation, meaning historical data can help predict future trends. The machine learning algorithms discussed earlier for predicting option prices frequently fail to account for the time series characteristics [5]. Consequently, this paper utilizes time series analysis techniques, such as Artificial Neural Networks (ANN) [17], Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Transformer, and Variational Quantized Variational Autoencoder (VQ-VAE) encoder, to improve the reliability of option price predictions [31]. RNNs, along with their variant LSTM, regulate the state transitions using a gating mechanism. However, they have certain limitations, including gradient vanishing and explosion, sequential processing, and a lack of long-term memory, which makes it difficult to handle long sequences of data. The Transformer model, on the other hand, can capture time series features from multiple time points through its self-attention mechanism and position encoding. It offers better long-term memory capabilities and enhanced performance in handling sequential data. Additionally, the VQ-VAE effectively learns feature representations from sequence data.

Additionally, as a financial derivative, an option's value is heavily influenced by the price of the underlying stock. Option pricing models, like the Black-Scholes model, also depend on factors such as stock price, volatility, and other market variables. [26] applied the gcForest algorithm for stock price prediction, achieving an annual return that exceeded the benchmark by 15.0%, with an excess return of 15.8%. The gcForest algorithm, which requires fewer hyperparameters, offers dis-

tinct advantages in stock price forecasting. In this study, we utilize the gcForest algorithm to predict stock prices. To address the overfitting problem, we propose an enhanced version of the ADgcForest algorithm, incorporating the three-objective optimization framework (TOOCEP) [22]. This approach prunes the deep forest hierarchically at the micro-pruning level, applying a cascade model to optimize each layer independently and select a subset. Subsequently, the deep forest is pruned layer by layer, from top to bottom. The proposed algorithm significantly reduces both storage requirements and prediction time.

Accurately predicting option prices is challenging due to the low signal-to-noise ratio, volatility, and computational complexity inherent in option data. To address the issue of the low signal-to-noise ratio, [20] introduces the R-measure method for option pricing, which effectively merges the actual measure with the risk-neutral measure, unifying them across different time periods. To enhance the accuracy of price predictions, we integrate the theoretical pricing model with a machine learning algorithm, utilizing prior knowledge to extract relevant characteristics from the option data. Consequently, we propose the R-factor, which combines the theoretical pricing model with AFgcForest, RNN, LSTM, Transformer, and VQ-VAE encoder, aiming to improve feature enhancement for option data. Through comparative experiments, we demonstrate that the proposed enhanced ADgcForest-R algorithm is capable of accurately predicting option prices over extended time series, while reducing computation time.

Based on a detailed analysis of the option market and pricing theory, this paper develops a feature-enhanced R-Optfa dataset and presents the ADgcForest algorithm, incorporating the TOOCEP three-objective pruning optimization technique, which demonstrates notable advantages in option price prediction. The proposed approach improves the ability to capture trends in the option market, thereby aiding in more effective investment decision-making.

The rest of our paper is organized as follows: Section 2 reviews related work, Section 3 introduces the enhanced gcForest algorithm and compares it with the R-measure algorithm, Section 4 outlines the datasets used in this study and explores the comparative experiments, and Section 5 concludes the paper.

2. Related Work

Machine learning algorithms, such as Self-Organizing Map (SOM), Support Vector Regression (SVR), and Support Vector Machine (SVM), are frequently used to analyze correlations in large stock data sets, demonstrating strong performance in stock price prediction [1]. [10] employs models like Candlestick Chart analysis, company fundamentals, and time series forecasting to predict stock prices across various markets. To enhance prediction accuracy, [9] applies the KNN algorithm to classify stocks based on their risk-return characteristics.

Many stock price prediction methods combine single or multiple behavioral asset models with Convolutional Neural Networks (CNN). These models predict the final

stock price. For example, [3] utilizes the DenseNet-41 algorithm to forecast closing prices based on the Yahoo Financial Data Stock Technical Index. Subsequently, numerous studies introduce innovations in model design and data usage to strengthen predictive performance. [2] proposes a Feedforward Deep Neural Network (FDNN) approach for effectively predicting abnormal stock prices in big data contexts. [16] applies the Neural Networks with Skip Connections model to precisely estimate the financial risk premium. There is also relevant work using Variational Autoencoders (VAEs) for stock price prediction. [27] employs VAE and Conditional Quantile VAE to model the conditional distribution of rewards in large-scale datasets for stock price prediction. [28] combines Gated Orthogonal Recurrent Units (GORU) with VAE to address the issues of gradient explosion and disappearance, thereby improving the accuracy and generalization of stock trend forecasting [7].

To improve the extraction of time series patterns in stock data, [25] developed a hybrid model that reduces prediction errors by incorporating machine learning techniques such as the Kalman filter, SVR, and nonlinear ANNs. LSTM network [19] performs exceptionally well in capturing the temporal dependencies of stock prices. [28] utilizes LSTM, based on asset pricing factors, to predict daily stock price trends and introduces a dynamic investment strategy that outperforms the benchmark index. To further enhance stock prediction accuracy, the MS-SSA-LSTM model [3], which integrates sentiment analysis and multi-source data, has demonstrated its ability to combine investor sentiment with fundamental trading data. Moreover, [8] optimizes LSTM using the Artificial Rabbits Optimization (ARO) method, improving the prediction accuracy for DJIA stocks.

The Transformer model is also effective for predicting stock prices. When analyzing stock market dynamics, its encoder-decoder structure and multi-head attention mechanism offer distinct advantages over other deep learning algorithms in predicting major global stock market indices [12]. The Teanet framework [4], which combines social media texts and stock prices, leverages the Transformer to predict stock trends with high accuracy [6].

The Teanet-framework [4], which integrates social media data and stock prices, leverages the Transformer model to accurately forecast stock trends and assess their practical value [6]. The Deep Transfer Metric Learning (DTML) approach utilizes a Transformer encoder to capture the relationships among various stocks, significantly enhancing investment simulation performance, including profit and annualized return [11]. Additionally, the Padding-based Fourier Transform Denoising (P-FTD) technique is introduced to efficiently remove noise from financial time series data [13]. The work in [15] further improves the Transformer by incorporating a multi-scale Gaussian prior, orthogonal regularization, and a trading gap separator algorithm, advancing the mining of long-term dependencies in financial time series. The T2V-TF model[29], which combines Time2Vec with Transformer for multi-source information fusion, boosts portfolio cumulative returns [12]. Together, these methods offer novel approaches and valuable insights.

3. Method

3.1. *R-measure Predictive Framework*

Extracting effective features from option data, which often contains a significant amount of non-critical information [20], is crucial. To address the issue of low signal-to-noise ratio in option data, we introduce prior knowledge by merging the theoretical model with the prediction model. This approach minimizes the search space and mitigates the risk of model over-fitting.

The R-measure is employed to incorporate the theoretical model into the stock prediction model. The pricing formula of the Black-Scholes-Merton (BS) Model under the R-measure is as follows [20]:

$$\mathbb{E}_t[S_H] = S_t e^{\mu(H-t)} = S_t e^{(1+f_3(X_t)+r_t)} \tag{1}$$

The R-Factors, computed under the R-measure as independent variables, are derived from prior economic knowledge. By integrating the prediction model, we embed the R-measure within this framework:

$$\begin{aligned} dY_s = &\left(r - q - \frac{v_s}{2} + 1_{\{s<H\}}\gamma^S v_s + 1_{\{s<H\}}\gamma^J\right) ds + \sqrt{v_s}\left(\rho dW_s^R + \sqrt{1-\rho^2} dW_{2s}^R\right) \\ &+ d\left(\sum_{i=1}^{N_s} J_{S,i}\right) - 1_{\{s<H\}}\lambda\bar{\mu} ds - 1_{\{s\geq H\}}\lambda^*\bar{\mu}^* ds, \\ dv_s = &\left[\alpha_v(m_v - v_s) - 1_{\{s\geq H\}}\gamma^v v_s\right] ds + \sigma_v\sqrt{v_s} dW_{1s}^R + d\left(\sum_{i=1}^{N_s} J_{v,i}\right). \end{aligned} \tag{2}$$

This framework leverages the strengths of the R-measure and computes the expected price of the index at any given future time. Subsequently, we calculate the expected excess return of the option with any maturity. Compared to the original framework, the integration of asset pricing theory and machine learning methods significantly enhances overall interpretability. This is because the mechanism of options is driven by deterministic features such as delivery price, term, and implied volatility. Simply using a large set of characteristic variables to predict the expected price of options does not guarantee that the trained model captures the theoretical relationship between these key characteristics. In the integrated framework, we focus solely on using the machine learning model to forecast the stock index, selecting only those variables relevant to stock index prediction as input. Since there is no defined theoretical connection between the characteristic variables of the stock index and those of the options, the model does not require theoretical validation. Then, based on the corresponding asset pricing model, the expected excess return can be derived. We define two R-factors: the expected theoretical price and the excess return of the option. The R-measure is applied to derive the formula for the expected return of the medium-term weight in the Merton model [20].

Assuming that the investment holding time $H(\leq T)$ and P conform to the standard Brownian motion model, the equivalent expected measure R also follows the standard Brownian motion form, as shown below.

$$W_s^R = W_s^P + \int_0^s \gamma 1_{\{u \geq b\}} du, \tag{3}$$

where γ represents the market price of risk, and $1\{\cdot\}$ is an indicator function that outputs one if the condition holds, and zero otherwise. This leads to the conclusion that the dynamics of $(A_t)_{t\geq 0}$ under the R-measure are described by

$$\frac{dA_s}{A_s} = \left(r + \sigma_A \gamma 1_{\{s<b\}}\right) ds + \sigma_A dW_s^R \tag{4}$$

$$= \left(\mu 1_{\{s<b\}} + r 1_{\{s\geq b\}}\right) ds + \sigma_A dW_s^R, \tag{5}$$

Where μ denotes the asset return drift under the P-measure. A key property of the R-measure is that it applies to any stochastic process governed by this measure. By inspecting the processes under both P and Q, the value process in Equation 4 can be derived. The stochastic process under R behaves as a physical process up until level H. However, it transitions to a risk-neutral process once H is reached or exceeded. It is important to note that when $H = 0$, the R-measure collapses to the Q-measure. The time-t expectation of the option at time $t + H$ (with $T = t$) is calculated as follows:

$$\mathbb{E}_t^P \left[D_{t+\tau}^T\right] = \mathbb{E}_t^R \left[e^{-r(\tau-b)} D_{t+\tau}^0\right] = \mathbb{E}_t^R \left[e^{-r(\tau-b)} \min(F, A_{t+\tau})\right] \tag{6}$$

Given that Equation 4 implies that $A_{t+\tau}$ follows a lognormal distribution under the R-measure, the price of a mid-horizon option in the Merton model is derived by the following equation.

$$\mathbb{E}_t^P \left[D_{t+\tau}^T\right] = F e^{-r(\tau-b)} N(d_2^b) + A_t e^{\mu b} N(-d_1^b), \tag{7}$$

$$d_1^b = \frac{\ln\left(\frac{A_t}{F}\right) + \mu b + r(\tau - b) + \frac{1}{2}\sigma_A^2 \tau}{\sigma_A \sqrt{\tau}}, \tag{8}$$

$$d_2^b = d_1^b - \sigma_A \sqrt{\tau}. \tag{9}$$

The expected option payoff, as indicated by the Merton model under P, can be computed using Equation 7. When $H = t$, this formula simplifies to the Black-Scholes-Merton model (BS). The expected excess return is then calculated based on the BS-implied expected price, which is designated as the BS-implied expected return. These two variables constitute the R-factor.

Building on this formulation of the R-factor, we propose a transfer learning framework for calculating the European option's expected price under the R-measure, as outlined below.

$$\mathbb{E}_t[C_H] = S_t e^{\mu(H-t)}\mathcal{N}(\hat{d}_1) - Ke^{-r(T-H)}\mathcal{N}(\hat{d}_2), \tag{10}$$

$$\hat{d}_1 = \frac{\ln\left(\frac{S_t}{K}\right) + \mu(H-t) + r(T-H) + \frac{1}{2}\sigma^2(T-t)}{\sigma\sqrt{T-t}}, \tag{11}$$

$$\hat{d}_2 = \frac{\ln\left(\frac{S_t}{K}\right) + \mu(H-t) + r(T-H) - \frac{1}{2}\sigma^2(T-t)}{\sigma\sqrt{T-t}}. \tag{12}$$

3.2. *ADgcForest with R-measure*

The gcForest [30] consists of two primary components: multi-grained scanning and the cascade forest structure. In multi-grained scanning, raw feature data is processed through sliding windows of varying sizes for the purpose of feature transformation and enhancement. On the other hand, the cascade forest structure is composed of multiple layers, including both a random forest and a complete random forest, enabling the model to automatically determine the number of layers without the need for manual hyperparameter design. This hierarchical setup facilitates the merging of features across different levels, improving both the complexity and performance of the model while keeping the number of hyperparameters to a minimum. The model's adaptability to datasets of varying sizes, coupled with its ability to prevent overfitting, strengthens its competitive edge.

The R-measure is integrated with the gcForest algorithm, with tri-objective optimization-based Single-Layer Pruning (TOOSLP) applied to prune and refine each layer. Additionally, it incorporates a new Chromosome Encoding Strategy developed using the Non-dominated Sorting Genetic Algorithm III (NSGA-III) evolutionary framework, accompanied by a fitness function that evaluates Accuracy, Independent Diversity, and Coupled Diversity. Furthermore, the cascade layer employs the pruning framework of Tri-Objective Optimization-Based Single-Layer Pruning (TOOSLP), thereby enhancing the overall efficiency of the pruning process.

The process of executing the Tri-Objective Optimization-based Single Layer Pruning (TOOSLP) algorithm in the gcForest model is outlined as follows: Initially, the cluster centers of the samples are extracted from the original dataset using the DBSCAN clustering algorithm, which serves as new features. Subsequently, Principal Components Analysis (PCA) is applied to perform feature selection and reduce dimensionality. Afterward, the gcForest model is trained using the reduced-dimensional features to generate an initial prediction. Next, the advanced Tri-Objective Optimization-based Single-Layer Pruning (TOOSLP) method is employed to conduct hierarchical post-pruning on the trained gcForest model. At each

level of the gcForest, the TOOCEP algorithm is applied to the decision tree set, optimizing the decision tree subset according to three objectives. In the first level, the decision tree set is encoded, and each decision tree is represented as a binary variable.

At the first level of the gcForest, the decision tree set is encoded, with each tree represented as a binary variable. A pool of candidate decision tree subset codes is then created, where each code represents a subset consisting of selected partial decision trees. Each subset code in the pool is evaluated based on the classification effect, prediction differences within the subset, and differences with previous decision trees, according to the three objective functions. The Non-dominated Sorting Genetic Algorithm III (NSGA-III) is employed to optimize the subset coding pool, and the Pareto optimal frontier solution for the first layer is determined. The subset code with the highest accuracy is selected as the result for the first-layer decision tree subset.

The integration of ADgcForest leverages the powerful feature transformation capabilities of gcForest, which excels in model complexity and abstraction, along with the efficient pruning abilities of TOOCEP. This synergy ensures that each layer of the cascaded forest not only enhances overall accuracy but is also optimized for greater efficiency, particularly in terms of storage and computational resources. Furthermore, TOOCEP's unique Coupled Diversity allows the pruning process to be customized for the specific cascade structure of gcForest, significantly reducing both prediction time and storage requirements. By combining the model complexity and abstraction strengths of gcForest with TOOCEP's three-objective pruning method, ADgcForest achieves superior prediction accuracy while enhancing efficiency in resource usage. As a result, ADgcForest demonstrates the advancement of ensemble learning and pruning techniques, offering an effective solution for complex learning problems.

The integration of the ADgcForest algorithm with the R factor, constructed through R-measure, enhances both the prediction accuracy and explanatory power of the model. As a component of R-measure theory, the R factor offers a unified framework for calculating the expected prices of financial products at various time points, thereby enriching the model with additional informative features. By incorporating the R factor alongside the ADgcForest algorithm, the prediction outcomes can be more effectively interpreted and understood.

3.3. *Transformer with R-measure*

The Transformer model adopts an encoder-decoder framework based on the attention mechanism, effectively replacing traditional recurrent neural networks and convolutional neural networks, thus achieving faster training times and enhanced performance. It utilizes components such as self-attention mechanisms, positional encoding, and feed-forward networks to capture global dependencies, extend its applicability to various tasks, and enhance interpretability. In machine translation and

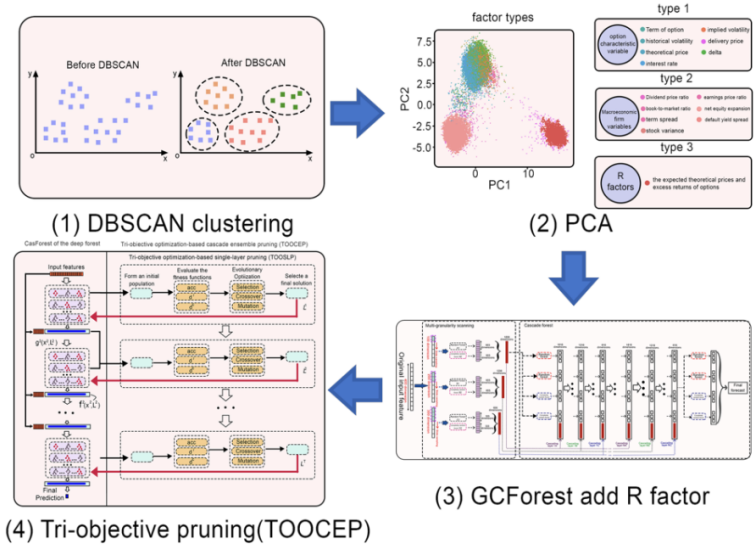

Fig. 1. ADgcForest Algorithm with R-measure)

other applications, the Transformer has shown strong performance, validating its significance and effectiveness within the domain of natural language processing [18]. This advanced data processing capability enables the Transformer model to detect subtle market signals, leading to improved accuracy and robustness in predicting option prices [14]. Typically, the final output layer includes one or more linear layers, which convert the processed data into specific predictions, such as the future price or return of options.

Incorporating R-measure option data into the Transformer model is designed to enhance both the performance and explanatory power of the model, particularly within the domain of financial analysis, with a focus on option pricing. Initially, the R-measure data must be encoded into a high-dimensional vector format that is compatible with the Transformer model. This encoding process preserves the multidimensional characteristics of each data point, which are then used as input to the model. To enable the model to effectively handle time series data, it is necessary to introduce location encoding into the input, which ensures that the temporal significance of the data is properly captured during processing.

The central component of the Transformer model is the Multi-Head Attention mechanism, which identifies complex dependencies between data by processing multiple attention heads simultaneously. By incorporating R-measure data, each attention head can not only capture conventional financial indicators but also analyze in-depth the potential relationship between risk and return as revealed by the R-

measure. The attention score for each head reflects the level of focus the model gives to input data at various time points when predicting option prices. This mechanism greatly improves the model's ability to understand the dynamics of the options market.

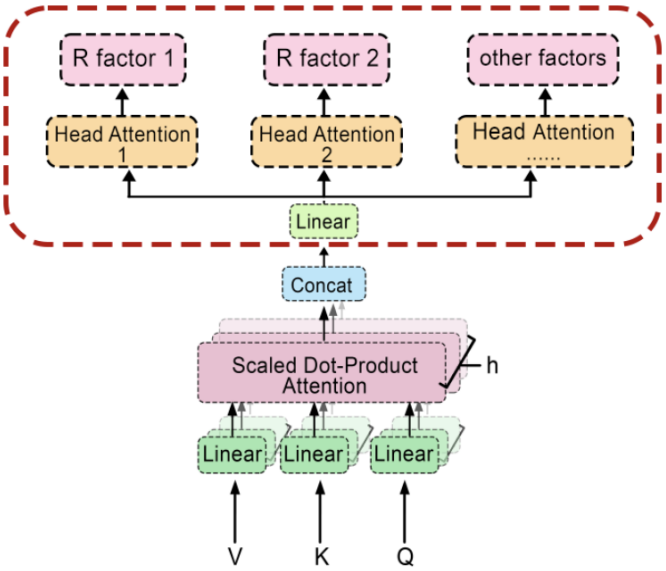

Fig. 2. Transformer Algorithm with R-measure

3.4. *VQ-VAE encoder with R-measure*

The Vector Quantized-Variational Autoencoder (VQ-VAE) model integrates Variational Autoencoder (VAE) with vector quantization, facilitating the extraction of essential features from data via discrete latent variables. In the context of finance, VQ-VAE is capable of identifying long-term dependencies within time series data, improving the efficiency of both data compression and generation, and performing unsupervised learning to uncover advanced patterns in financial data. These attributes offer substantial support for the analysis and prediction of financial trends. This research explores the incorporation of the R-measure into the VQ-VAE model to improve the accuracy of option pricing and optimize risk evaluation.

In the encoding phase of the VQ-VAE model, by introducing a risk adjustment layer based on the R-measure, we adjust the encoder to capture the features related to market risk. This layer emphasizes the risk-sensitive aspects of option data, and by modifying feature weights, the model directs more attention to key risk factors when analyzing and processing financial data. In addition, the R-measure enables the encoder to generate more comprehensive and detailed feature representations,

better capturing the dynamics of the market and the underlying risk framework.

In the R-measure feature learning layer, the codebook vector is refined to align with the risk evaluation criterion established by the R measure, thus improving the learning mechanism of the VQ-VAE model. This refinement guarantees that the model's encoding not only captures the overall characteristics of the data but also effectively represents the risk-benefit framework. Consequently, the model is able to reconstruct and predict option prices with greater accuracy during the decoding phase, particularly in high-risk market environments.

In the decoder, we train it to focus on encoding vectors with high sensitivity to risk. This approach allows the decoder to prioritize key risk factors that could trigger market volatility when reconstructing option prices, thereby enhancing the model's application value and prediction accuracy in real-world financial settings. As a result, the decoder can produce more accurate market forecasts at the output stage, assisting financial analysts in making more informed and robust investment decisions.

During the final stage of model training, we refine the loss function by incorporating a risk adjustment factor based on the R-measure, ensuring that the model can effectively enhance its predictive accuracy for risk while minimizing reconstruction errors. This approach not only boosts the model's performance in risk-sensitive markets but also improves its generalization capability, allowing it to better adapt to and predict changes in the market environment. These targeted improvements demonstrate the effectiveness of combining the R-measure with the VQ-VAE model, offering new insights and methodologies for applying deep learning techniques to the financial sector in the future.

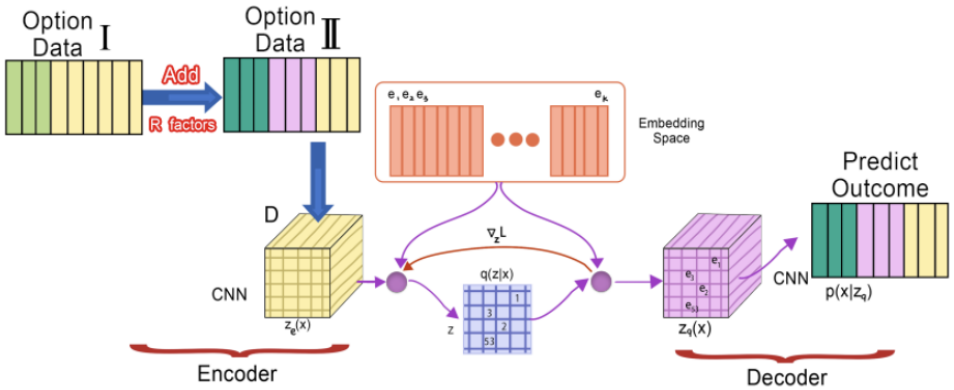

Fig. 3. VQ-VAE Algorithm with R-measure

4. Experiments

4.1. *Dataset Information*

We create a comprehensive feature dataset by collecting historical data on Standard & Poor's 500 Index Options from 1998 to 2021. The features include factors such as the option's expiration date, implied volatility, historical volatility, implied strike price, implied premium, Black-Scholes-Merton (BS) model theoretical price, risk-free interest rate, delta value, and closing price on the contract signing date. Traditionally, some asset returns, such as those from options, are difficult to capture in a factor model using economically interpretable factors. Building on the predicted option return performance [21], we introduce seven stock characteristics: Dividend-Price Ratio, Earnings-Price Ratio, Book-to-Market Ratio, Net Equity Expansion, Term Spread, Default Yield Spread, and Stock Variance. From these 18 variables, two R factors are constructed using a structured model. The first factor is derived from the Black-Scholes Implied Expected Price to examine the Merton model's effect on cross-sectional returns and their time-series predictability. After pre-processing operations, such as normalization, missing value imputation, and extreme value replacement, we create the Option-Factor dataset, named R-Optfa.

Tables 1 and 2 present the descriptive statistics for the R-Optfa dataset. A recursive method is employed for model validation, where the R-Optfa dataset is divided into three subsets: training, validation, and testing. This approach helps identify any data structure instability and allows for evaluating the model's prediction performance across different time periods. Table 1 provides statistical information on returns and option characteristics from 1998 to 2021. Option returns are calculated over a period preceding the option's expiration. The BS-implied expected return, one of the R factors, is computed under the R index to determine the excess return. The expiration date is measured as the number of calendar days remaining until the option expires. The option value is defined as the ratio of the stock price to the option's strike price. Implied volatility is sourced from Option Metrics. Panels B and C present statistics for call and put options, respectively.

4.2. *Performance comparison with gcForests*

To evaluate the predictive power of the gcForest algorithm, we conducted experiments without incorporating R factors on the training, validation, and test sets. The R^2 values obtained were 0.9981, 0.9786, and 0.9536, respectively, while the Root Mean Square Error (RMSE) values were 0.0606, 0.1997, and 0.2756. These results suggest that the gcForest algorithm achieves optimal performance on the training set, maintains strong accuracy on the validation set, and experiences a slight decline on the test set, reflecting its overall excellent predictive ability.

The enhanced ADgcForest algorithm, evaluated without R factors, achieved R^2 values of 0.9997, 0.9988, and 0.9978 on the training, validation, and test sets, respectively. The corresponding RMSE values were 0.0239, 0.0467, and 0.0606. In

Table 1. Summary Statistics for Options (Part 1)

	Mean	Sd	Min	Q10	Q25	Median
A: All Options: ($n = 2,108,371$)						
Realized Excess Return	-0.23	1.15	-1.14	-1.02	-1.00	-0.99
BS-implied Expected Return	0.86	3.14	-1.13	-1.00	-0.83	-0.13
Days to Maturity	238.12	214.89	10.00	30.00	91.00	152.00
Moneyness	1.01	0.13	0.41	0.86	0.94	1.01
Implied Volatility	0.20	0.07	0.03	0.12	0.14	0.19
Historical Volatility	0.17	0.09	0.02	0.11	0.11	0.15
Delta	0.00	0.56	-0.90	-0.75	-0.50	-0.10
B: Call Options: ($n = 1,041,048$)						
Realized Excess Return	0.04	1.22	-1.14	-1.02	-1.00	-0.20
BS-implied Expected Return	1.69	3.99	-1.13	-0.98	-0.56	0.37
Days to Maturity	234.43	212.65	10.00	30.00	91.00	152.00
Moneyness	1.00	0.11	0.44	0.85	0.94	1.00
Implied Volatility	0.20	0.07	0.05	0.12	0.14	0.19
Historical Volatility	0.17	0.09	0.02	0.11	0.11	0.15
Delta	0.51	0.24	0.10	0.15	0.30	0.50
C: Put Options: ($n = 1,067,323$)						
Realized Excess Return	-0.49	1.01	-1.14	-1.02	-1.01	-0.38
BS-implied Expected Return	0.05	1.65	-1.13	-1.00	-0.92	-0.51
Days to Maturity	241.72	217.00	10.00	30.00	91.00	152.00
Moneyness	1.01	0.13	0.51	0.86	0.95	1.01
Implied Volatility	0.20	0.07	0.06	0.12	0.14	0.19
Historical Volatility	0.17	0.09	0.02	0.11	0.11	0.15
Delta	-0.50	0.24	-0.90	-0.85	-0.70	-0.50

comparison to gcForest, ADgcForest demonstrates notably higher R^2 values across all datasets, especially improving the test set R^2 from 0.9536 to 0.9978 and reducing RMSE from 0.2756 to 0.0606. These improvements are primarily attributed to enhanced data preprocessing techniques and a hierarchical tri-objective pruning strategy, which were validated through comprehensive experimentation.

With the integration of R factors, the ADgcForest algorithm achieved even higher R^2 values of 0.9998, 0.9992, and 0.9987 for the training, validation, and test sets, respectively, along with RMSE values of 0.0227, 0.0380, and 0.0457. This improvement further highlights the effectiveness of R factors in enhancing the model's predictive power, particularly on the test set, where R^2 rose from 0.9978 to 0.9987 and RMSE dropped from 0.0606 to 0.0457. The incorporation of R-measure theory into the gcForest model significantly boosts its expressive capacity, enabling it to make more precise predictions of option prices by capturing and utilizing the

Table 2. Summary Statistics for Options (Part 2)

	Q75	Q90	Max	Skew	Kurt
A: All Options: ($n = 2,108,371$)					
Realized Excess Return	0.35	1.30	5.85	1.86	3.94
BS-implied Expected Return	1.10	3.49	26.95	3.86	19.46
Days to Maturity	365.00	547.00	730.00	1.10	0.07
Moneyness	1.06	1.14	3.05	0.49	3.95
Implied Volatility	0.23	0.29	1.28	1.43	4.59
Historical Volatility	0.21	0.27	1.37	2.45	10.76
Delta	0.50	0.75	0.90	0.01	-1.37
B: Call Options: ($n = 1,041,048$)					
Realized Excess Return	0.70	1.59	5.85	1.41	2.37
BS-implied Expected Return	2.02	5.77	26.95	3.03	10.86
Days to Maturity	365.00	547.00	730.00	1.14	0.17
Moneyness	1.07	1.13	1.97	0.62	2.57
Implied Volatility	0.23	0.29	1.28	1.44	4.67
Historical Volatility	0.21	0.27	1.37	2.47	11.01
Delta	0.70	0.85	0.90	-0.04	-1.18
C: Put Options: ($n = 1,067,323$)					
Realized Excess Return	0.84	1.80	5.85	2.65	8.05
BS-implied Expected Return	0.33	1.72	26.85	4.13	29.27
Days to Maturity	365.00	547.00	730.00	1.07	-0.02
Moneyness	1.07	1.16	3.05	0.84	3.84
Implied Volatility	0.23	0.29	1.21	1.42	5.04
Historical Volatility	0.21	0.27	1.37	2.42	10.82
Delta	-0.30	-0.15	-0.10	0.01	-1.28

intrinsic relationships between option prices at different time points.

When tested on Google Colaboratory, the ADgcForest algorithm demonstrated superior efficiency compared to the gcForest algorithm, being approximately 0.15 seconds faster without R factors and about 0.12 seconds quicker with R factors. This suggests that ADgcForest excels not only in predictive accuracy but also in operational efficiency, a critical aspect for real-time option trading.

In conclusion, whether utilized with or without R factors, ADgcForest shows notable improvements in both predictive performance and efficiency. These advantages stem from enhanced data preprocessing and a hierarchical tri-objective post-pruning approach. Furthermore, other models incorporating R-measure factors and macroeconomic company variables failed to match the performance of ADgcForest, highlighting its potential for practical application in machine learning-based option trading. The performance boost can be attributed to the R-measure factors,

which offer a unified framework for estimating the expected prices of financial products, thereby enriching the model's feature set and improving its forward pricing capabilities.

Table 3. Performance comparison with gcForests

Table 2: gcForest results

Index\Group		Training Set		Validation Set		Test Set		
		R^2	RMSE	R^2	RMSE	R^2	RMSE	Testing Time
No R Factor	gcForest	0.9981	0.0606	0.9786	0.1997	0.9536	0.2756	4.4759s
	ADgcForest	0.9997	0.0239	0.9988	0.0467	0.9978	0.0606	4.3243s
R Factor	ADgcForest	0.9998	0.0227	0.9992	0.0380	0.9987	0.0457	4.3552s

4.3. *Performance comparison with State-of-the-arts*

To evaluate the predictive performance of various time series modeling methods that incorporate R factors, we are conducting experiments on the training, validation, and test datasets.

The RNN model yields R^2 values of 0.6579, 0.6553, and 0.6567 for the training, validation, and test sets, respectively, with corresponding RMSE values of 0.6725, 0.6728, and 0.6731. These results indicate comparable performance on the training and validation sets, but a slight drop in the test set, suggesting that the model has difficulty capturing long-term, but a slight decrease in accuracy on the test set, suggesting challenges in capturing long-term dependencies.

The LSTM model, incorporating R factors, achieves R^2 values of 0.7873, 0.7829, and 0.7847 for the training, validation, and test sets, respectively, with corresponding RMSE values of 0.5303, 0.5339, and 0.5330. When compared to the RNN model, LSTM shows better performance across all datasets, reflected in lower RMSE values, which suggests its enhanced ability to capture long-term dependencies.

Following parameter optimization, the Transformer model attains R^2 values of 0.8245, 0.6757, and 0.7989 for the training, validation, and test sets, respectively, along with RMSE values of 0.5982, 0.7646, and 0.6473. Notably, the test set R^2 improves from -0.002 to 0.7989, and RMSE decreases from 1.4456 to 0.6473, underscoring the significance of optimization. Despite this progress, the Transformer still lags behind the gcForest model, likely due to gcForest's superior handling of time series patterns and long-term dependencies.

The VQ-VAE encoder, incorporating R factors, achieves R^2 values of 0.8590, 0.8543, and 0.8534 for the training, validation, and test sets, respectively, with corresponding RMSE values of 0.4318, 0.4374, and 0.4398. These results demonstrate the encoder's strong ability to extract features and compress data; however, it still trails behind gcForest in overall performance.

The ADgcForest model, also including R factors, delivers the best results with R^2 values of 0.9998, 0.9992, and 0.9987 across the training, validation, and test sets, respectively, and RMSE values of 0.0227, 0.0380, and 0.0457. These findings underscore ADgcForest's superior capability in handling time series data and performing predictive tasks.

In terms of testing time, the LSTM model performs the fastest, followed by the optimized Transformer, the unoptimized Transformer, and the RNN model. While LSTM benefits from a speed advantage, its predictive performance falls short compared to gcForest and ADgcForest. On the other hand, the VQ-VAE model is the slowest, with the unoptimized Transformer, optimized Transformer and LSTM. RNN models ranking faster. While VQ-VAE excels in feature extraction, its processing speed may create a bottleneck.

In conclusion, both the RNN and LSTM models struggle to effectively manage long-term dependencies, while the optimized Transformer shows some improvement, yet still underperforms relative to gcForest and ADgcForest. The VQ-VAE encoder offers strong feature extraction capabilities, but its overall performance is hindered by slower processing speeds when compared to gcForest. The latter's ensemble structure and multi-tree fusion allow it to capture complex data relationships more efficiently. Notably, the ADgcForest model stands out for its superior predictive accuracy and time efficiency, making it a highly effective tool for machine learning-based option trading predictions.

Table 4. Performance comparison with State-of-the-arts on R-Optfa dataset

Table 3: Results of the model with R-factors

Index\Group		Training Set		Validation Set		Test Set		
		R^2	RMSE	R^2	RMSE	R^2	RMSE	Testing Time
R Factor	RNN	0.6579	0.6725	0.6553	0.6728	0.6567	0.6731	5.1224s
	LSTM	0.7873	0.5303	0.7829	0.5339	0.7847	0.5330	4.2550s
	Transformer	0.8245	0.5982	0.6757	0.7646	0.7989	0.6473	4.9634s
	VQ-VAE	0.8590	0.4318	0.8543	0.4374	0.8534	0.4398	6.2377s
	ADgcForest	0.9998	0.0227	0.9992	0.0380	0.9987	0.0457	4.3552s

5. Conclusion

This study evaluates various algorithms for machine learning-based option trading prediction. Our results show that the gcForest algorithm performs consistently well without R factors. However, the enhanced ADgcForest algorithm, which incorporates clustering, PCA, and tri-objective pruning, outperforms gcForest both with and without R factors. In contrast, RNN and LSTM models face challenges in managing long-term dependencies. Despite improvements from optimization, the

Transformer model still underperforms relative to gcForest. While the VQ-VAE encoder excels in feature extraction, its overall performance lags behind gcForest. Notably, ADgcForest demonstrates exceptional efficiency, a crucial advantage in the fast-paced trading environment. The limitations of RNN and LSTM models in handling dependencies, coupled with their longer testing times, may restrict their practical use. In conclusion, ADgcForest stands out for its strong predictive accuracy and processing efficiency, making it an ideal choice for machine learning-based option trading prediction.

Looking ahead, we plan to expand our research by exploring dynamic feature selection to further improve both predictive accuracy and efficiency. Our focus will be on enhancing ADgcForest using dynamic programming techniques. Additionally, we aim to apply ADgcForest to other financial instruments, such as futures and swaps, in order to develop a more comprehensive machine learning-based trading framework. Through these efforts, we hope to strengthen the robustness and versatility of our algorithm in the ever-changing landscape of financial markets.

Funding Statement

This work is not supported by any funding.

References

1. A. Aijaz, K. Rastogi and D. T. Sivakumar, Stock market analysis and prediction using machine learning, *International Journal of Recent Technology and Engineering (IJRTE)* **11**(2) (2022) 54–60.
2. T. Al-Sulaiman, Predicting reactions to anomalies in stock movements using a feed-forward deep learning network, *International journal of information management data insights* **2** (2022) p. 100071.
3. S. Albahli, A. Awan, T. Nazir, A. Irtaza, A. Alkhalifah and W. Albattah, A deep learning method dcwr with hanet for stock market prediction using news articles, *Complex & Intelligent Systems* **8** (2022) 2471 – 2487.
4. S. Albahli, T. Nazir, M. Nawaz and A. Irtaza, An improved densenet model for prediction of stock market using stock technical indicators, *Expert systems with applications* **232** (2023) p. 120903.
5. S. Chen, J. Qiu, H. Zhang, Y. Yu, H. Chen and Y. Sun, Speech fatigue recognition under small samples based on generative adversarial networks and blstm, *International Journal of Pattern Recognition and Artificial Intelligence* **38**(13) (2024) p. 2458005.
6. M. Guarascio, M. Minici, F. S. Pisani, E. D. Francesco and P. Lambardi, Movie tag prediction: An extreme multi-label multi-modal transformer-based solution with explanation, *Journal of intelligent information systems* (2024) 1–23.
7. H. Gunduz, An efficient stock market prediction model using hybrid feature reduction method based on variational autoencoders and recursive feature elimination, *Financial Innovation* **7** (2021).
8. B. Gülmez, Stock price prediction with optimized deep lstm network with artificial rabbits optimization algorithm, *Expert systems with applications* **227** (2023) p. 120346.
9. A. Kowalska, Study on using machine learning-driven classification for analysis of the disparities between categorized learning outcomes, *Electronics* **11**(22) (2022) p. 3652.

10. D. Lade, A. Patil, P. Yenkar, S. Alone and S. Dhawas, Stock market prediction using machine learning, *International journal of advanced research in computer and communication engineering* **10** (2023) p. 2717.
11. J. Leng, W. Liu and Q. Guo, Stock movement prediction model based on gated orthogonal recurrent units, *Intelligent systems with applications* **16** (2022) p. 200156.
12. S. Li. and H. Wu, Transformer-based denoising adversarial variational entity resolution, *Journal of intelligent information systems* **61**(2) (2023) 631–650.
13. F. Lin, P. Li, Y.-C. A. Lin, Z. Chen, H. You and S. Feng, Kernel-based hybrid interpretable transformer for high-frequency stock movement prediction, *2022 IEEE International Conference on Data Mining (ICDM)* (2022) 241–250.
14. F. Liu, D. Yang, Y. Zhang, C. Yang and J. Yang, Research on multi-parameter prediction of rabbit housing environment based on transformer, *Int. J. Data Warehous. Min.* **20** (2024) 1–19.
15. Y. Liu, X. Xu, G. Trajcevski and F. Zhou, Transformer-enhanced hawkes process with decoupling training for information cascade prediction, *Knowledge Based Systems* **255** (2022) p. 109740.
16. A. W. Lo and M. Singh, Deep-learning models for forecasting financial risk premia and their interpretations, *Quantitative Finance* **23**(6) (2023) p. 917–929.
17. B. Madhu, M. A. Rahman, A. Mukherjee, M. Z. Islam, R. Roy and L. E. Ali, A comparative study of support vector machine and artificial neural network for option price prediction, *Journal of Computational Chemistry* **36** (2021) p. 301121.
18. N. Mamta and A. Ekbal, Transformer based multilingual joint learning framework for code-mixed and english sentiment analysis, *Journal of intelligent information systems* **62**(1) (2023) 231–253.
19. E. Mantel, An international study of application of long short-term memory (lstm) neural networks for the prediction of stock and forex markets, *International Journal For Multidisciplinary Research* **5**(3) (2023).
20. S. Nawalkha and X. Zhuo, A theory of equivalent expectation measures for expected prices of contingent claims, *SSRN Electronic Journal* **77**(5) (2020) 2853–2906.
21. S. Palaiahnakote, D. Kapri, M. H. Saleem and U. Pal, A new contrastive learning-based vision transformer for sentiment analysis using scene text images, *International Journal of Pattern Recognition and Artificial Intelligence* (2024).
22. R. Ran, T. Gao, Q. Hu, W. Zhang, S. Peng and B. Fang, Representation learning based on vision transformer, *International Journal of Pattern Recognition and Artificial Intelligence* **38**(07) (2024) p. 2459004.
23. S.-R. Stefano, V. Gissel and Z. Damian, A machine learning approach for bitcoin forecasting, *Engineering Proceedings* (2023).
24. N. Umeorah, P. Mashele and J. Mba, Barrier options and greeks: Modeling with neural networks, *Axioms* **12** (2023) p. 384.
25. C. Venkatachalam, M. Umamaheswari, P. Shah and A. Thakur, Revolutionizing brain tumor diagnosis: A comprehensive model integrating vgg19 and lstm for accurate mri classification, *International Journal of Pattern Recognition and Artificial Intelligence* (2025) p. 2457015.
26. Q. Xu and J. Guo, Multi-target detection method of intelligent driving traffic scene based on faster r-cnn++, *International Journal of Pattern Recognition and Artificial Intelligence* (2025) p. 2459019.
27. X. Yang, Z. Zhu, D. Li and K. Zhu, Asset pricing via the conditional quantile variational autoencoder, *Journal of Business & Economic Statistics* **42**(2) (2023) p. 681–694.
28. H. Yao, S. Xia and H.-C. Liu, Six-factor asset pricing and portfolio investment via

deep learning: Evidence from chinese stock market, *Pacific-basin Finance Journal* **76** (2022).

29. F. Zhou, Q. Zhang, Y. bing Zhu and T. Li, T2v-tf: An adaptive timing encoding mechanism based transformer with multi-source heterogeneous information fusion for portfolio management: A case of the chinese a50 stocks, *Expert systems with applications* **213** (2022) p. 119020.
30. Z.-H. Zhou and J. Feng, Deep forest: Towards an alternative to deep neural networks, *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence* (2017) 3553–3559.
31. Q. Zhu, C. Wang, W. Jin, J. Ren and X. Yu, Deep transfer learning based on lstm model for reservoir flood forecasting, *Int. J. Data Warehous. Min.* **20** (2024) 1–17.