IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 June 2024, accepted 22 July 2024, date of publication 29 July 2024, date of current version 6 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3434528

RESEARCH ARTICLE

A Deep Learning Based Expert Framework for Portfolio Prediction and Forecasting

FATHE JERIBI[1], R. JOHN MARTIN[1], RUCHI MITTAL[2], HASSAN JARI[1], ABDULRAHMAN HASSAN ALHAZMI[1], VARUN MALIK[2], (Member, IEEE), S. L. SWAPNA[3], S. B. GOYAL[4], (Senior Member, IEEE), MANOJ KUMAR[5,6], (Member, IEEE), AND SHUBHRANSHU VIKRAM SINGH[7], (Senior Member, IEEE)

[1]College of Engineering and Computer Science, Jazan University, Jazan 45142, Saudi Arabia
[2]Chitkara Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab 140401, India
[3]Applexus Technologies, TechnoPark, Thiruvananthapuram 695583, India
[4]Faculty of Information Technology, City University Malaysia, Petaling Jaya 46100, Malaysia
[5]School of Computer Science, FEIS, University of Wollongong in Dubai, Dubai Knowledge Park, Dubai, United Arab Emirates
[6]MEU Research Unit, Middle East University, Amman 11831, Jordan
[7]Amity University, Noida, Uttar Pradesh 201301, India

Corresponding author: R. John Martin (jmartin@jazanu.edu.sa)

This work was supported by the Deanship of Graduate Studies and Scientific Research, Jazan University, Saudi Arabia, under Project GSSRD-24.

ABSTRACT Stock market forecasting involves predicting fluctuations and trends in the value of financial assets, utilizing statistical and machine learning models to analyze historical market data for insights into future behavior. This practice aids investors, traders, financial institutions, and governments in making informed decisions, managing risks, and assessing economic conditions. Forecasting financial markets is difficult due to the intricate interplay of global economics, politics, and investor sentiment, making it inherently unpredictable. This study introduces a Deep Learning based Expert Framework for Stock Market forecasting (Portfolio prediction) called DLEF-SM. The methodology begins with an improved jellyfish-induced filtering (IJF-F) technique for preprocessing, effectively analyzing raw data and eliminating artifacts. To address imbalanced data and enhance data quality, pre-trained convolutional neural network (CNN) architectures, VGGFace2 and ResNet-50, are used for feature extraction. Additionally, an improved black widow optimization (IBWO) algorithm is designed for feature selection, reducing data dimensionality and preventing under-fitting. For precise stock market predictions, integrate deep reinforcement learning with artificial neural network (DRL-ANN) is proposed. Simulation outcomes reveal that the proposed framework achieves maximum forecasting accuracy, reaching 99.562%, 98.235%, and 98.825% for S&P500-S, S&P500-L, and DAX markets, respectively.

INDEX TERMS Stock market, predictive analytics, portfolio management, deep learning, feature optimization.

I. INTRODUCTION

Predicting future trends and price movements in the stock market necessitates the use of a number of different analytical methods [1]. Forecasting the stock market helps traders and investors plan their stock purchases and sales in light of future market circumstances. Predicting the movement of the stock market [2] is important because it enables buyers and sellers of stocks to better manage their portfolios. The high volatility of stock prices is due to the many variables that might impact them, including market movements, economic data, political events, corporate news, and investor mood. Forecasting helps investors and traders to anticipate the direction of stock prices, identify potential risks or opportunities, and make well-informed investment decisions. Furthermore, stock market forecasting is crucial for financial institutions, such as banks and insurance companies, as they need to anticipate market trends and movements in order to manage risk and make profitable investments [3], [4]. Machine learning and deep learning techniques are gaining traction in stock market

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Ni.

forecasting due to their capacity to analyze vast datasets and uncover intricate patterns and relationships, surpassing traditional methods [5]. Research and development efforts are focused on these cutting-edge methods because they have the potential to greatly enhance the reliability of stock market predictions. Moreover, precise stock market predictions can assist investors in recognizing potentially lucrative opportunities and maximizing their investment returns. Furthermore, stock market forecasts [6] can help businesses to make informed decisions about their operations, such as expansion, product development, and marketing campaigns, based on expected market conditions.

There are several traditional techniques used for stock market forecasting, fundamental analysis, technical analysis, economic indicators and expert opinions [7]. On the other hand, technical analysis entails examining historical stock price and volume data, alongside other market indicators like moving averages, to detect trends and patterns that aid in forecasting future price movements [8], [9]. Combining these conventional techniques [10] is common practice, offering a more holistic perspective of the market and aiding investors in making informed decisions regarding stock purchases and sales. As machine learning (ML) and deep leaning (DL) [11], [12] can process massive volumes of data and spot intricate patterns in that data, they are finding growing utility in stock market predictions. Time-series analysis is a method for predicting the future price of company by evaluating its price movements in the past. Financial data may be analyzed using deep learning methods like convolutional neural networks (CNNs) [13] and recurrent neural networks (RNNs) [14] to reveal hidden patterns and correlations.

The volatility of financial time series has been predicted using a PSOQRNN (particle swarm optimization-trained quantile regression neural network) [15]. For the purpose of forecasting stock returns, a deep learning-based model was created [16] that employ principal component analysis, auto-encoder, and limited Boltzmann machine to build three-layer artificial neural networks. Macroeconomic factors and stock market linkages have also been studied using structural models [17]. For stock market forecasting specifically, an improved multi-factor and Type-2 fuzzy time series model has been presented [18]. Stock market volatility may now be predicted using a one-step-ahead model that syndicates empirical mode decomposition and a stochastic time-strength neural network [19]. In addition, the stock market's decision-making processes may now be bolstered by an autonomous emotional decision-making system [20].

An efficient deep learning based expert framework is proposed for stock market forecasting using feature optimization (DLEF-SM). The major contributions of our proposed DLEF-SM framework are list as follows:

- An improved jellyfish-induced filtering (IJF-F) technique is used for preprocessing the raw data and removing artifacts.
- The pre-trained architectures, VGGFace2 and ResNet-50 is used for the feature extraction which is used to address the issue of imbalanced data and improve data quality.
- An improved black widow optimization (IBWO) algorithm is further for feature selection to reduce data dimensionality and prevent under-fitting.
- For stock market forecasting, we utilize the deep reinforcement learning with artificial neural networks (DRL-ANN), which guarantees high detection accuracy.
- The proposed DLEF-SM framework is validated by benchmark datasets, including S&P500-S, S&P500-L, and DAX market.

The remaining sections of this work will be structured as follows. In the second part, we'll look at some of the current research and developments in the field of experts-based stock market forecasting. In Section III, we provide the DLEF-SM framework's issue approach and system design. In Section IV, a mathematical model is used to describe how the proposed DLEF-SM framework operates in detail. Part 5 discusses the simulation results and the comparative analysis of the proposed and existing frameworks. In Section 6, the paper comes to a close.

II. RELATED WORKS

The works included under Related Work give a survey of what has already been written on stock market forecasting. Table 1 provides a synopsis of the knowledge gaps in the field.

Using deep reinforcement learning (DRL), Li et al. [21] created efficient stock transaction strategies and proved DRL's applicability when dealing with financial strategy difficulties. They assessed efficacy of three conventional DRL models and discovered that DQN model performed better when it came to stock market strategy decision-making challenges. Ding et al. [22] were able to predict stock market returns. The method outperformed both linear and nonlinear autoregressive models in identifying differences between mature and developing markets. The models showed the promise of AI-based techniques in stock market forecasting by achieving an improvement rate of roughly 30% for in-sample fitting and 40% for out-of-sample forecasting.

Lee et al. [23] proposed robust framework for stock market prediction. In order to regularize the model, the scientists used data augmentation method to convolutional neural networks. The mean squared error reductions for S&P500, KOSPI200, and FTSE100 of NuNet were 60.79%, 51.29%, and 43.36%, respectively.

Lyocsa et al. [24] offer a long-term stock market prediction model for predicting the realized variance of key market indexes. Performance gains for their model on literature-standard benchmarks range from 6.57% (SSEC) to 35.62% (NIKKEI 225) for the MSE loss function and from 3.99% (STOXX) to 9.54% (NIKKEI 225) for the QLIKE loss function.

TABLE 1. Summary of research gaps.

Ref	Methodology	Feature extraction	Forecasting model	Dataset	Findings	Research gaps
[21]	Stock trading forecasting	League champion	Deep Q learning	ETF, NASDAQ	MACD, KAMA	Training with large datasets can take a significant amount of time.
[22]	Forecasting stock market return	Autoregressive	Genetic programming	WIND	MAE, MSE	Small changes in the training data can lead to unstable results.
[23]	Stock market forecasting	CNN and Max-pooling layer	NuNet	S&P500, KOSPI200	MAE, MSE	Computational resources may incur significant costs, particularly when dealing with sizable datasets.
[24]	Stock market volatility forecasts	HAR	HAR-CSLR and HAR-CSQR	NIKKEI 225, S&P 500, STOXX 50	MAE, MSE	The number of clusters must be predetermined, which can be challenging.
[25]	Forecasting stock index price	SVM	CEEMDAN-LSTM	S&P500	MAE, MAPE, MSE, RMSE	The algorithm is sensitive to the local structure of the data, which can affect its performance.
[26]	Intraday stock price forecasting	NA	ANN models	SBIN.NS, INFY.NS	Accuracy 97.24%,	The convergence rate of the algorithm can be slow.
[27]	Forecasting daily stock trend	L1–LR, SVM, RF	MFFS	88 NASDAQ listed stocks	Accuracy 59.44%	The training process can be affected by local minima.
[28]	Stock market prediction on oil and gas sector	Auditory algorithm	SVM, LR, FNN	Nigerian stock exchange (2014-19)	MAE, MAPE, MSE, RMSE	Unequal time series data lengths pose challenges in determining weight vector scales, impacting performance.
[29]	Forecasting stock price	BA and GA	ANN	S&P500, DAX, FTSE100	MAE, MAPE, MSE, RMSE	The selection of parameters can greatly influence the performance of the algorithm.
[30]	Forecasting US stock price	NA	WT-ANFIS	NASDAQ100	MAE, MAPE	Training can take longer than decision algorithms.

Lin et al. [25] introduced the CEEMDAN-LSTM model for stock index price forecasting. The model's predicting accuracy and resilience in both established and developing stock markets were evaluated and confirmed using an MCS test. Chandar [26] proposed nine integrated models for intraday stock price forecasting. PSO-BPNN demonstrated the most favorable outcomes across various stocks, even when training and testing samples were interchanged. Haq et al. [27] established model for daily stock trend predictions. A deep generative approach is used for the forecasting future price movements. The deep generative model uses a market signal extractor and an attention mechanism to interpret hidden variables in market movements and discern predictive patterns among various temporal auxiliary outputs. Oyewola et al. [28] introduced the auditory algorithm (AA), which takes its cues from the natural world. They used statistical tests to evaluate the AA algorithm's performance to that of six other algorithms.

Farahani et al. [29] devised a method to forecast stock price indices by training artificial neural networks (ANNs) with metaheuristic algorithms such as social spider optimization (SSO) and bat algorithm (BA). Sharma et al. [30] offer a forecasting model that combines ANFIS and Wavelet Transform. The empirical findings show that the WT-ANFIS model with a trapezoidal MF performs better than the model with a bell-shaped MF.

A. RESEARCH GAPS

Thus, several theories and strategies have been presented for predicting stock prices. Optimization techniques such as neural networks and decision trees sit alongside more conventional statistical approaches like the autoregressive integrated moving average (ARIMA) model. Yet, each approach has benefits and drawbacks, and picking the right one requires considering the details of the data and the issue at hand. In addition, newer and more complex models are continuously generated and evaluated for accuracy and dependability with the use of deep learning and other developing technologies. To help anticipate the unpredictable and chaotic behavior of stock markets, Carta et al. [31] present a set of Deep Q-learning classifiers. The approach employs ensemble framework that incorporates diverse agreement criteria and data resolutions, along with multiple agents possessing varying degrees of market expertise. Forecasting models also be useful for companies to predict their financial performance, plan their budget, and make strategic decisions [34]. In general, forecasting models can help individuals and organizations to anticipate future outcomes and take proactive measures to achieve their goals. There are several problems associated with deep learning-based stock market forecasting models. Deep learning models can be very complex and have a large number of parameters, which makes them prone to overfitting if they are not properly regularized.

While deep learning models can make accurate predictions, it can be difficult to interpret the reasons behind those predictions, which can be problematic for investors who need to understand the underlying factors driving stock price movements. When high-quality financial data is scarce and deep learning models need massive volumes of data for training, it may be difficult to construct reliable models. The statistical features of stock market data often shift over time, making the data non-stationary [40]. In the development of a stock market prediction model, feature extraction stands out as one of the most crucial tasks [41]. Extracting useful characteristics from raw data for use in making accurate stock price predictions is the goal here. The problem of feature extraction arises due to the large number of potential features that can be extracted from the stock market data, making it challenging to select the most informative ones [42]. Moreover, the relevance of the features may vary with time, and new features may become relevant over time. Therefore, there is a need for effective feature selection methods that can adapt to changes in the data and select the informative features for accurate forecasting. The problem in feature optimization arises because large number of potential features available for use, and selecting the most important ones is crucial for accurate forecasting [43]. However, selecting too many features lead to under-fitting or overfitting of the model, respectively, resulting in poor performance. Based on the problems in stock market forecasting, some possible research objectives are:

- To develop new feature extraction methods that are robust, efficient and effective in capturing relevant information from different types of data sources.
- To find ways to optimize and choose attributes that are most important for stock market forecasting so that the models may be more accurate and applicable in general.
- To develop DL models that surpasses traditional ML models by addressing the challenges posed by non-linearity, non-stationary, and high dimensionality in stock market data.
- The goal is to measure and analyze how well the suggested forecasting models perform on various benchmark datasets, comparing their results to those of state-of-the-art methodologies.

III. MATERIALS AND METHODS

Here, we describes how the suggested framework which employs cutting-edge methods and algorithms for data preprocessing, feature extraction and selection, and forecasting actually works to enhance the reliability of stock market predictions.

A. SYSTEM ARCHITECTURE OF THE PROPOSED FRAMEWORK

The stock market forecasting system that we propose to use deep learning and efficient feature optimization (DLEF-SM) is shown in Figure 1. The dataset used in this study includes three stock market indices: SP500-S, SP500-L, and DAX. These indices are widely used as indicators of the performance of the US and German stock markets. To remove noise from the stock market data, three filters are used: wavelet transforms (WT), singular spectrum analysis (SSA), and Kalman filter. These filters are applied to the data to reduce the impact of random fluctuations and other unwanted signals. The IJF-F technique is used to preprocess the data. This technique is based on the idea of removing the high-frequency components from the data and retaining only the low-frequency components. This aids in enhancing the forecasting model's accuracy, potentially bolstering its performance through the mitigation of noise and outlier effects. From the cleaned and prepared data, features are extracted using two deep learning models: VGGFace2 and ResNet-50. Although ResNet-50 is often used for image identification applications, VGGFace2 is a convolutional neural network (CNN) developed specifically for face recognition. This method may be useful for spotting indicators in stock market data that may foretell future price movements. With the help of the improved black widow optimization (IBWO)

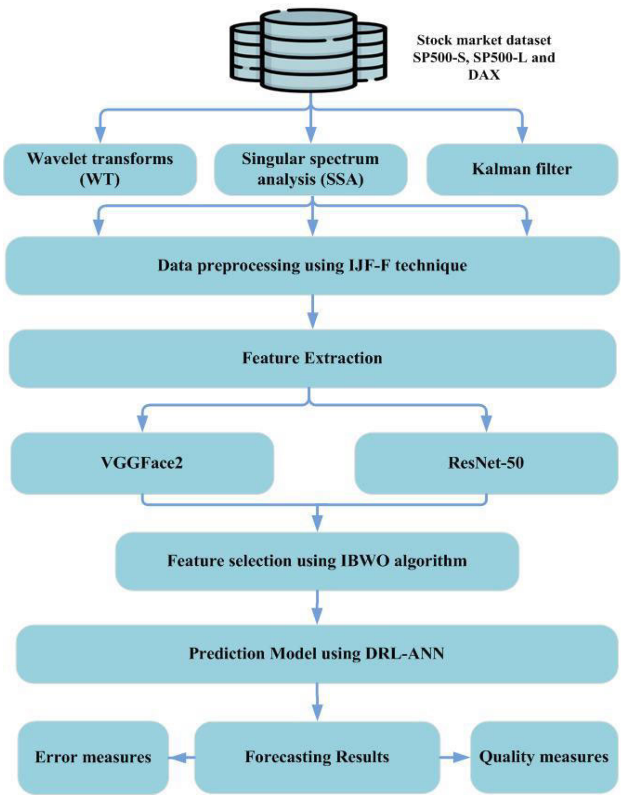


FIGURE 1. Overall system architecture of proposed DLEF-SM framework.

algorithm, we can zero down on the data points that will prove most useful to our prediction model. For this purpose, the system employs a binary whale optimization technique to zero in on the best possible collection of characteristics to utilize in its prediction process. The dataset's dimensionality may be decreased in this way, making it more manageable for further processing and analysis. For stock market forecasting, experts turn to the deep reinforcement learning-artificial neural network (DRL-ANN) method. DRL-ANN is a hybrid model that combines the strengths of deep reinforcement learning (DRL) and artificial neural networks (ANN) to better understand and forecast data. Quality metrics such as accuracy, precision, recall, and F1-score, along with error metrics like MAE, MSE, and RMSE, are commonly employed to evaluate forecasts. By utilizing these metrics, analysts can assess the effectiveness of the forecasting model and compare it with other models currently employed.

B. DATA PREPROCESSING USING IMPROVED JELLYFISH-INDUCED FILTERING (IJF-F) TECHNIQUE

Preparing the data for analysis is what data preprocessing in stock market forecasting is all about. In addition, data normalization and scaling are used to make the data uniform and comparable. During preprocessing, characteristics may be chosen such that only the most useful ones are sent into the prediction model. In last, the cleansed and sorted data is partitioned into a training set and a testing set for use in developing and testing the model's forecasting capabilities. As a whole, data preprocessing is an essential stage in stock market forecasting since it has such a direct bearing on the reliability and precision of the forecasting outcomes. In this study, well-established methods are used for data preparation in stock market forecasting. These methods include wavelet transformations (WT) [31], singular spectrum analysis (SSA) [31], and the Kalman filter [32]. It is possible to remove noise and extract useful features from a time series by using wavelet transformations, which break the series into scales and frequencies. Time series decomposition may also be accomplished by singular spectrum analysis, which involves splitting the original signal up into its spectral components. Kalman filter is a statistical technique used for data smoothing, where noisy or erratic data points are removed and replaced with predicted values based on the surrounding data. Overall, data preprocessing techniques are crucial in stock market forecasting as they help to remove noise and identify important features, making it easier for forecasting models to identify trends and make accurate predictions. IJF-F is a metaheuristic optimization algorithm that takes cues from the fish-finding techniques of jellyfish. IJF-F aims to improve the search performance and convergence speed of JOA by incorporating a new mutation strategy and a dynamic parameter adaptation mechanism. The mutation strategy involves the addition of a random vector to the current solution vector to enhance the diversity of the search space. The dynamic parameter adaptation mechanism adjusts the mutation and crossover rates based on the convergence state of the algorithm to balance exploration and exploitation.

Each example in the training set, which has m samples $\{(x^{(j)}, y^{(j)})\}$, j = 1…m, has a feature vector $x^{(j)}$ and a corresponding label $y^{(j)}$. We compute DCNN $h\{z, A\}$ with weights z and bias A, the model output $\tilde{y}^{(j)}$ of the model sample $(x^{(j)}, y^{(j)})$ can be calculated. By presenting the loss function, an error is obtained. Loss function B usually has cumulative error and control period is compute as follows.

$$I(z, A) \approx \frac{1}{N} \sum_{j=1}^{N} L(h\{z, A\}(x^{(j)}, y^{(j)})) + \lambda \sum_{j,i} z_{j,i}^2 \quad (1)$$

where $I(z, A)$ stands for the input model error, with N being the volume size. The magnitude and setup of relative errors are within the control of the hyperparameter λ. The discrepancy between the two values is reported in square meters.

$$C = \frac{1}{2n} \sum_{x} \|y(x) - B(x)\|^2 \quad (2)$$

In determining the slope, the default coefficient of 1/2 wipes out the effect of factor 2.

$$A^{(N+1)} = A^{(N)} - \frac{\eta}{N} * \Delta A \quad (3)$$

where $A^{(N)}$ is the shift at the N^{th} cycle with N representing the overall sample count. By assigning the same weight (η) to all of the nodes in the same layer, MJSO simplifies the process of configuring the network and opens up more options for handling massive amounts of data.

$$X_{+1} = \eta X_j(1 - X_j),\ \ 0 \leq X_0 \leq 1 \tag{4}$$

The j[th] jellyfish's logistically chaotic standards are included in the vector X_j. Jellyfish 0 is the first element of a vector X_0 that may be generated freely between 0 and 1. We may use this vector as a springboard to generate illogical values for jellyfish. Each jellyfish's current location is tracked and updated by a system that regulates its transitions between groves, ocean currents, or both. We compute the optimal solution $X_j(s+1)$ as follows.

$$X_j(s + 1) = X_j(s) + r. * (x^* - \beta * R_1 * \mu) \tag{5}$$

The vector r consists of values that are close to both zero and one. The vector growth ($*$) occurs one element at a time, μ is the population average, and r1 is a random integer between 0 and 1, $\beta > 0$ is the allocation coefficient. The jellyfish swim aimlessly about their current locations before being relocated according to the following formula:

$$X_j(s + 1) = X_j(s) + R_3 * \gamma * (V_a - L_a) \tag{6}$$

where γ a random number ranging from 0 to 1is included in the vector R. The vector R contains random numbers between 0 and 1. A jellyfish's offspring may be predicted to swim in a certain direction by looking at their D. The temporal mechanism that carries both passive and active motion, and which is formally described as:

$$C(t) = (1 - \frac{T}{T_{\max}}) * (2 * R - 1) \tag{7}$$

The primary goal of fitness that employs the described approach for addressing the parameter extraction issue is to find values for the unknown parameters that minimize the difference between the optimal and predicted in the data. Current and the simulated current are used to calculate the root mean squared error (RMSE) of the optimal fitness. The following is a detailed description of the optimal fitness that should be used along with the specification.

$$RMSE = F(X_j) = \sqrt{\frac{1}{m} * \sum_{K=1}^{m} (j_m - j_E(U, X_j))^2} \tag{8}$$

Calculated and expected currents, j_m and j_e, are shown. The length of the data set under consideration is represented by m. The jth solution's predictable parameters are denoted by x_j. Utilizing the parameters represented in x_j and the Newton-Raphson, we may determine j_e.

$$j_e = j_e - \frac{df}{df'} \tag{9}$$

where df is $J's$ step-wise solution. Using J as the derivational axis, df yields df' first.

$$df = j_{qh} - j_{td}(\exp(p(U + j * r_s) - 1) - \frac{U + j * R_t}{r_{th}} - 1 \tag{10}$$

$$df' = -j_{td}\frac{pr_t}{NKSn_t}(\exp(\frac{U + j * R_t}{NKSn_t}) - 1) - \frac{R_t}{r_{th}} - 1 \tag{11}$$

The optimal solution for the best search range, R, effectively computes the control process of preprocessed data and is increased as maximum range even though R is a modest and comprehensive inquiry. The formulas for the pre-integration technique are:

$$\begin{aligned} X_j(s + 1) &= X_j(s) + R * (X_{R1}(s) - x_{R2}(s)) \\ &\quad + (1 - R) * (X^* - X_{R3}(s)) \end{aligned} \tag{12}$$

where R is the regulating variable, and R1, R2, and R3 are the indices of the solutions drawn at random from the population. Algorithm 1 outlines the operational steps of data pre-processing employing the Improved Jellyfish-Induced Filtering (IJF-F) technique.

Algorithm 1 Data pre-processing using IJF-F technique

Input : weights z
Output: control variables

1. Start by Initializing the random population
2. Set initial fitness using
   $C = \frac{1}{2N} \sum_x \|y(x) - B(x)\|^2$
3. Compute ocean current as
   $X_j(s + 1) = X_j(s) + r. * (x^* - \beta * R_1 * \mu)$
4. If j =0 and i =1
5. Compute $C(t)$ as $C(t) = (1 - \frac{T}{T_{\max}}) * (2 * R - 1)$
6 Induct the pre-integration method using
   $X_j(s + 1) = X_j(s) + R * (X_{R1}(s) - x_{R2}(s)) + (1 - R) * (X^* - X_{R3}(s))$
7. Return the final best solution
8. End

C. FEATURE EXTRACTION USING VGGFACE2 AND RESNET-50

Feature extraction is a process of selecting relevant and important features or attributes from raw data in order to facilitate data analysis and machine learning algorithms. The raw data may contain many features that are not useful or redundant for the task at hand, and feature extraction aims to transform the data into a more compact representation that retains the most relevant information for the task. In stock market forecasting, feature extraction is vital as it converts raw data into pertinent and valuable features, capturing the underlying patterns and dynamics of the stock market. To prepare data for analysis and modeling, one must first identify and then choose or extract the most interesting and relevant characteristics from the raw data. Following feature selection, the identified characteristics are employed to train machine learning models, which will then forecast future trends and behaviors in the stock market. In order to make more precise and efficient stock market predictions, effective feature extraction algorithms are essential. In this research, we present the idea of feature extraction and discuss its application to stock market forecasting by implementing it using VGGFace2 and ResNet-50.

1. VGGFace2 is a popular deep learning model for doing facial recognition. But, it may be used to feature extraction in other fields, such as financial market prediction. High-level features may be extracted from photos using the VGGFace2 model, which has already been trained on a huge dataset of faces. For more precise predictions, the VGGFace2 model is often used in tandem with other deep learning models like the long short-term memory (LSTM) model [24]. The VGGFace2 model is helpful for complicated datasets like stock market data due to its ability to extract a high number of characteristics from photos. Image identification and object detection are only two of the areas where it has been demonstrated to excel, and research into its potential application in stock market predictions is ongoing.
2. A popular choice for image identification applications is ResNet-50, a deep convolutional neural network design. In stock market forecasting, ResNet-50 can be used for feature extraction from the time series data. Each stock market dataset (SP500-S, SP500-L, and DAX) can be treated as a 1D image, with time as the horizontal axis and the stock market value as the vertical axis. The ResNet-50 architecture consists of several convolutional layers with residual connections, which allow for better gradient flow and help to avoid the vanishing gradient problem. During feature extraction, each input time series is passed through the ResNet-50 network, and the activations of the last layer are extracted as features. These characteristics are then sent into the feature selection phase, where only the most relevant ones are chosen to be utilized in the prediction model.

D. FEATURE SELECTION USING IMPROVED BLACK WIDOW OPTIMIZATION (IBWO) ALGORITHM

Feature selection often comes after feature extraction. In order to increase the prediction model's accuracy and efficiency, it is common practice to "feature pick," or choose a subset of attributes that are most relevant from the whole set of available features. Feature selection aids in stock market forecasting by highlighting the most important qualities that are expected to have a major influence on company prices. The predictive model's precision and safety against overfitting may both benefit from this. It was suggested as a step up from the regular black widow optimization (BWO) method. IBWO introduces a new mechanism to handle the exploration-exploitation tradeoff by considering the diversity of the population in addition to the fitness values of candidate solutions [35]. This is achieved by using a diversity-based selection strategy that promotes the selection of diverse solutions in the population. In addition, IBWO employs an adaptive mutation strategy that enhances the algorithm's ability to escape from local optima. IBWO has demonstrated encouraging outcomes in addressing diverse optimization tasks, such as feature selection, image classification, and power system optimization. Both the quality of its solutions and the pace at which it converges to them has been shown to be superior to those of previous meta-heuristic optimization techniques.

The IBWO, on the other hand, replicates the bizarre mating of black widow spiders. During the generation phase, create an extensive window array filled with random integers and designate it as alpha (α).Alpha (α) serves as a parent, producing offspring, with resulting individuals having alpha (α) and other parents. The crossover outcome is then assessed and documented.

$$x_1 = \alpha \times y_1 + (1 - \alpha) \times y_2 \tag{13}$$
$$x_2 = \alpha \times y_2 + (1 - \alpha) \times y_1 \tag{14}$$

The window size, represented by n_{var}, signifies the ideal solution to an optimization problem with n_{var} dimensions.

$$Window = [y_1, y_2, \ldots .y_{n_{var}}] \tag{15}$$

The data utilized for the variable $[y_1, y_2, \ldots .y_{n_{var}}]$ are considered as decimals. Window fitness is determined by put on the fitness function f to a set of $[y_1, y_2, \ldots .y_{n_{var}}]$ windows.

$$Fitness = F(Window) = F[y_1, y_2, \ldots .y_{n_{var}}] \tag{16}$$

By iterating in this manner $n_{var}/2$ times, it guarantees that no two randomly selected numbers are identical. Finally, the entire family unit is assembled into an array and sorted based on fitness, now considering the cannibalism rating, and the top individuals are retained for the subsequent generation. IBWO streamlines the search's expansion phase by randomly selecting indices instead of altering all significant factors of the population's position, as done in the original BWO.

$$\lambda = \lambda_{\max} \cdot \exp(\log \frac{\lambda_{\min}}{\lambda_{\max}}) \cdot \frac{Iter}{Iter_{\max}} \tag{17}$$

In this context, λ represents the fruit fly search range for the ongoing iteration, $\lambda_{\max}$ for the maximum, and $\lambda_{\min}$ for the minimum. The ongoing iteration is symbolized by $Iter$, while $Iter_{max}$ represents the maximum permitted iteration.

$$y_{j,i} = \begin{cases} \delta_i \pm \lambda \cdot rand() & if\ \ i = D \\ \delta_i & otherwise \end{cases} \quad , i = 1,\ 2,\ \ldots N \tag{18}$$

The position $y_{j,i}$ is updated as δ_i the best possible solution value in the i^{th} dimension, D denotes an index randomly selected from uniformly distributed choice variables, while N indicates the dimension of the solution, and *rand()* denotes a random integer within the range [0,1]. The initial positions of the flies can be randomly selected.

$$Y_{best} = qx\ \ rand\_val\ (domain\ definition) \tag{19}$$
$$Y_j = \omega x\ (domain\ definition) x\ rand\_val(-1, 1) \tag{20}$$

Here, ω represents the IBWO search engine. The process of determining candidate selections based on volunteer scores establishes the link between volunteer evaluations and the resultant potential choices for each search. When it moves towards the top individual in IBWO, relocate it to their position.

$$Y(s+1) = Y^{*}(s) + d_q E^{al} \cos(2\pi l) \tag{21}$$

where, E^{al} remains constant, dictating the shape of the spiral, while l takes on a random value within the interval $[-1, 1]$. $d_q = |Y^{*}(s) - Y(s)|$ denotes the difference between the ideal factor Y prior to update and the optimal position Y_{best}.

$$d = cY_{rand} - Y(s) \tag{22}$$

$$Y(s+1) = Y_{rand} - B \times d \tag{23}$$

Selecting a whale's location at Y_{rand}, at random. The phases of the feature selection algorithm employing IBWO are outlined in Algorithm 2.

Algorithm 2 Feature selection using IBWO

Input: well known and unknown features, termination condition

Output: best optimal feature

1. Initiate the random population
2. Set the y_1 and y_2 are parents, x_1 and x_2 are descendants
3. If j =0 and i =1
4. Adjust the position upon reaching the optimal individual during swimming $Y(s+1) = Y^{*}(s) + d_q E^{al} \cos(2\pi l)$
5. The random number between [−1,1]
6. Return the final potential solution
7. End

E. STOCK MARKET FORECASTING USING DRL-ANN

Forecasting the future movement of stock prices and stock market indexes is known as "stock market forecasting," and it involves the use of a wide range of analytical and statistical methods. Methods for forecasting future stock market movements based on an examination of historical trends, patterns, and other variables with bearing on stock prices. Investment planning, risk management, and portfolio optimization are just some of the many uses for stock market forecasting. It's a difficult job that requires in-depth knowledge of financial markets and the use of cutting-edge analytical tools and methodologies. DRL-ANN uses DRL to optimize the ANN model by learning from past market data and adjusting its parameters accordingly [36]. It can handle the high-dimensional and non-linear nature of financial data, which makes it an ideal tool for stock market forecasting. By using DRL-ANN, the predictive model can continuously learn from new data and adapt to changing market conditions, making it an effective tool for traders and investors to make informed decisions. Standard reinforcement learning methods involve an agent interacting with its environment (here, denoted as e) within discrete time intervals, aiming to maximize long-term rewards to acquire optimal control strategies. At each time step, the reinforcement learning agent observes the state t_s of e and selects an action to maximize anticipated future rewards. The environment e subsequently responds by providing the agent R_s with the next state $R_s + 1$ and a scalar reward.

$$r_s = \sum_{K=0}^{\infty} R^K R_s + K \tag{24}$$

In state s, if you do the action a, you may expect to get $B(t, p)$ as a reward. Maximum potential profit equals $B^{*}(t, p)$, where

$$B^{*}(t, p) = \max e[r_s | t_s = t, \; p_s = p] \tag{25}$$

Bellman's equation holds true for the B-function. In order to choose the best course of action, one must maximize the anticipated value of r, where t and p represent the state and action at the next time step, respectively.

$$B^{*}(t, p) = Max\, e[R + \gamma \; \max \; P^{*}(t', p') \,|t, p| \tag{26}$$

In general, the reinforcement learning algorithm optimizes its control approach by teaching itself via repeated, data-driven interactions with the environment.

The policy network π furnishes a probability distribution of favorable actions based on the current state st, while the value network $v_{\theta V}(t_s)$ evaluates the expected return from state t_s using the Bellman equation, akin to the Q-function in a conventional reinforcement learning framework, with parameters θ and θ_V weights, respectively.

$$l_V(\theta_V) = e\,(r_s - v_{\theta V}(t_s))^2 \tag{27}$$

By the use of value-based approaches, agents are trained to make optimum estimates of a value function, which then determines the agent's policy by selecting the most valuable action. The function of the state value is often defined as follows.

$$V^{\pi}(t) = e\;[r(\tau)|t] \tag{28}$$

To compute the value function of states and actions as follows.

$$P^{\pi}(t, p) = e\;[r(\tau)|t, p] \tag{29}$$

where and represent the cumulative discounted benefit anticipated from state t onwards. Then, go down path τ in accordance with policy π. It's not hard to see how these two ideas are connected:

$$V^{\pi}(t) = e\;[B^{\pi}(t, p)] \tag{30}$$

In other words, $V^{\pi}(t)$ is the sum of $B^{\pi}(t, p)$ divided by the probabilities of each action taken. The issue takes on a fixed shape if and only if $M_t(N, J)$ tunable parameters indicate a test solution in terms of q.

$$\min_{j} \sum_{\overrightarrow{N_b \in d}} f(N_b, M_T(N_b, J), M_T'(N_b, J)) \tag{31}$$

It dependents on the limitations set by BCs. The suggested method uses DRL-ANN for the trial solution Y, with parameters p representing the neural architecture's weights and biases. For many purposes, ANNs are the commonly used model because of its "black box" nature. When it comes to hydro-environmental engineering, ANN models have been widely used as a time-series prediction tool. In order to derive the exact equation that describes how an ANN arrives at its output value, one must first

$$\hat{M}_a = f_a[\sum_{H=1}^{A} W_{aH} \times f_H(\sum_{b=1}^{b} W_{bH} N_b + W_{Hx}) + W_{ax}] \quad (32)$$

where f_H, f_b signify activation function for the hidden and output layers; b, H, a, p, and W represent neurons in the input, hidden, and output layers; and the weight and bias imposed by the neuron. Controller variations were ablated in one experiment. The cosine similarity between an observed sample and memory keys forms the basis of our reading controller. Here's how we get at the previous read similarity s π and the context read similarity t_{γ}^{b}:

$$t_{\pi}^{j} = \frac{\pi^K \cdot \pi^b}{||\pi^K||\ ||\pi^b||} \qquad j = 0, \ldots.., |p| \quad (33)$$

$$t_{\gamma}^{b} = \frac{\gamma^K \cdot \gamma^b}{||\gamma^K||\ ||\gamma^b||} \qquad b = 0, \ldots.., |p| \quad (34)$$

After this, we use t_{π}^{j} and t_{γ}^{q} as inputs to a multilayer feed-forward neural network F, which integrates read similarities, weight history, and contextual significance, trained to assign high scores to relevant samples and low scores to others. As a result, we can calculate the overall read probability by doing the following:

$$J(r)^b = f(t_{\pi}^{b}, t_{\gamma}^{b}) \quad (35)$$

Given that we can read and decode each memory sample separately, all we need to do to achieve multimodality is read the top-K samples with the largest $J(r)^b$ during inference. The solution is to express it as the sum of two words. To create this term, a DRL-ANN whose weights and biases need to be changed to solve the minimization issue may be used. The following is the output of the DRL-ANN given the input y.

$$q = \sum_{b=1}^{H} v_b \sigma(w_b), \qquad where\ w_b = \sum_{a=1}^{q} w_{ba} N_a + x_b \quad (36)$$

Here, w_{ba} the input unit b signifies the load connecting the hidden unit to b, v_b the input unit q represents the load connecting b to the output unit, and x_b the hidden unit embodies the dependence of b and $\sigma(w)$. Additionally, σ is a sigmoidal transfer function. Algorithm 3 describes the step involved in the Stock market forecasting using DRL-ANN.

Algorithm 3 Stock Market Forecasting Using DRL-ANN

Input : optimal best features, training set and testing set
Output: stock market forecasted results

1 Set the values for the input parameters
2 Compute cumulative reward using
$r_s = \sum_{K=0}^{\infty} R^K R_s + K$
3 Define B-function in typical reinforcement learning setup using
$l_V(\theta_V) = e(r_s - v_{\theta V}(t_s))^2$
4 While do
5 The output value of an ANN is compute using
$\hat{M}_a = f_a[\sum_{H=1}^{A} W_{aH} \times f_H(\sum_{b=1}^{b} W_{bH} N_b + W_{Hx}) + W_{ax}]$
7 Return the final values
8 End if

IV. RESULTS AND DISCUSSIONS

This section presents the simulation findings and analysis of both the current and newly proposed stock market forecasting systems. We begin by briefly describing the experimental setup used for the stock market forecasting. This includes details such as the dataset used, data preprocessing techniques applied, feature extraction and selection methods used, and the forecasting model used. After describing the experimental setup, we present the results obtained in a clear and concise manner, including appropriate tables, figures, and statistical measures to support our findings. This should include a comparison of the different forecasting models used, highlighting their respective strengths and weaknesses. We evaluated the proposed DLEF-SM framework on benchmark datasets, such as S&P500-S, S&P500-L, and DAX. The framework was written in Python and tested on a 64-bit Windows 10 PC running on an Intel i7-8700k, 64GB of RAM and an Nvidia TITAN XP GPU using the Google Colab simulation environment. We compare the proposed DLEF-SM framework's simulated performance to that of well-established, industry-standard frameworks like random forest (RF), decision tree (DT), logistic regression (LR), support vector machine (SVM), deep neural network (DNN), long short term memory (LSTM), convolutional neural network (CNN), double Q-learning (DQL), and multi-DQN [31].

A. DATASET DESCRIPTION

Several datasets, including the S&P500 and the DAX, are used to assess the effectiveness of newly proposed and the most recent forecasting systems (DAX). Here, we split the S&P500 dataset in half and run two sets of experiments: one on a subset of data spanning only two years (which we've dubbed SP500-S) and the other on the whole dataset, which spans a full decade (we call this sub-dataset SP500-L). Experiments used 22 epochs of SP500-L data, 5 epochs of SP500-S data, and 21 epochs of DAX data (although the time periods of SP500-L and DAX are the same, there are missing days for DAX that causes the reduced number of epochs). We used five years of training data, six months of validation data, and six months of testing data for each time period. Assume, for the sake of argument, that we have

TABLE 2. Dataset description.

Description	Datasets		
	SP500-S	SP500-L	DAX
Data duration	2015-2017	2007-2017	2007-2017
Number of entries	929	3557	2643
Number of long entries	508	1946	1409
Number of short entries	421	1611	1234
Buy-and-Hold strategy return (USD)	27,537.5	52,725	193,275
Number of training samples	744	2845	2115
Number of testing samples	185	712	528

a 200-day dataset that we want to split into six 100-day walks. The first 80 days of each epoch might be used as a training set, while the latter 20 days could be used as a testing set. Table 2 provides a description of the datasets used in the study. The SP500-S, SP500-L, and DAX datasets were utilized. The data duration for each dataset varied, with SP500-S covering the period from 2015 to 2017, SP500-L covering the period from 2007 to 2017, and DAX covering the period from 2007 to 2017. The table also shows the number of entries in each dataset, which varied from 929 for SP500-S to 3557 for SP500-L. Additionally, the table indicates the number of long and short entries in each dataset. For instance, SP500-S had 508 long entries and 421 short entries. In addition, the table displays the USD buy-and-hold strategy return for each dataset; in this case, the SP500-S saw a return of 27,537.5, the SP500-L saw a return of 52,725, and the DAX saw a return of 193,275. Lastly, the table shows how many samples were utilized for each phase of the study's training and testing phases. Therefore, SP500-S, for instance, used 744 training samples and 185 testing samples.

B. RESULT ANALYSIS WITH RESPECT TO PREPROCESSING TECHNIQUES

The outcomes of the proposed framework for the SP500-S dataset using various pretreatment filtering strategies are shown in Table 3. Some of the forecasting frameworks taken into account include RF, DT, LR, SVM, DNN, LSTM, CNN, DQN, Multi-DQN, and the proposed DLEF-SM. Mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE) are the error measurements often compared (MAE). Wavelet transformations (WT), singular spectrum analysis (SSA), and the Kalman filter are the preparatory filtering methods used. It is clear by comparing the MSE, RMSE, and MAE values that DLEF-SM outperformed Multi-DQN, DQN, and CNN. Using the Kalman filter as a preprocessing technique, the models' performance showed an increase in accuracy compared to using the WT and SSA filters.

For the SP500-S dataset, the proposed DLEF-SM framework with wavelet transform (WT) preprocessing outperforms existing frameworks in terms of all error measures. Specifically, the MSE of DLEF-SM is 2.449%, 2.183%, 1.915%, 1.646%, 1.375%, 1.103%, 0.83%, 0.555%, and 0.278% lower than the existing RF, DT, LR, SVM, DNN, LSTM, CNN, DQN, and Multi-DQN frameworks, respectively. The RMSE and MAE of DLEF-SM with WT preprocessing also demonstrate significant improvement compared to existing frameworks. For SSA preprocessing, DLEF-SM achieves the lowest RMSE and MAE among all frameworks, but its MSE is slightly higher than that of the existing RF and DT frameworks. The MAE of DLEF-SM is 2.332%, 2.06%, 1.786%, 1.514%, 1.241%, 0.969%, 0.696%, 0.424% and 0.151% less than the existing RF, DT, LR, SVM, DNN, LSTM, CNN, DQN, and Multi-DQN frameworks, respectively. For the Kalman filter preprocessing, DLEF-SM outperforms existing frameworks in terms of the RMSE and MAE measures, but its MSE is slightly higher than that of the existing RF and DT frameworks. The RMSE of DLEF-SM is 4.3%, 4.034%, 3.767%, 3.501%, 3.234%, 2.968%, 2.701%, 2.435% and 2.169% lower than the existing RF, DT, LR, SVM, DNN, LSTM, CNN, DQN, and Multi-DQN frameworks, respectively. Overall, the DLEF-SM framework consistently outperforms the existing frameworks across all error measures and forecasting models, with the largest improvements observed for the IJF-F-Kalman filter.

Referring to the radar plots shown in Figure 2, Figure 3, and Figure 4 for the datasets SP500-S, SP500-L, and DAX dataset respectively, a clear wreck in the top left part of the plots indicate that the proposed model exhibits elevated performance over all other models for all three preprocessing techniques namely IJF-F-WT, IJF-F-SSA and IJF-F-Kalman.

Based on Table 4, the proposed framework (IJF-F) with different preprocessing filtering techniques (WT, SSA, and Kalman) has been compared with various forecasting frameworks based on error measures such as MSE, RMSE, and MAE for the SP500-L dataset. We find that the suggested DLEF-SM framework significantly reduces the MSE, RMSE, and MAE compared to the state-of-the-art frameworks. In particular, DLEF-MSE SM's values are lower than those of the other frameworks, suggesting that it performs better in terms of accuracy. In addition, DLEF-RMSE SM's and MAE values are lower than those of the competing frameworks, demonstrating it does a better job of forecasting the SP500-L dataset's actual values. As a result, it seems that the suggested structure can withstand a wide variety of preprocessing filters. Looking at Table 5, it is clear that the suggested IJF-F-WT framework outperforms the other models in terms of MSE, RMSE, and MAE on the DAX dataset. With an MSE of 0.491, the DLEF-SM model outperforms the next-best model, the Multi-DQN model, by 1.2%. The DLEF-SM

TABLE 3. Results of proposed framework with different preprocessing filtering techniques for SP500-S dataset.

Forecasting frameworks	Error measures								
	MSE			RMSE			MAE		
	IJF-F-WT	IJF-F-SSA	IJF-F-Kalman	IJF-F-WT	IJF-F-SSA	IJF-F-Kalman	IJF-F-WT	IJF-F-SSA	IJF-F-Kalman
RF	0.478	0.503	0.475	0.691	0.716	0.659	0.478	0.503	0.495
DT	0.477	0.502	0.474	0.679	0.704	0.606	0.466	0.491	0.483
LR	0.475	0.500	0.473	0.667	0.692	0.634	0.453	0.478	0.451
SVM	0.474	0.499	0.471	0.654	0.679	0.652	0.441	0.466	0.418
DNN	0.473	0.498	0.47	0.642	0.667	0.636	0.429	0.454	0.406
LSTM	0.471	0.496	0.469	0.630	0.655	0.627	0.416	0.441	0.401
CNN	0.470	0.495	0.468	0.617	0.642	0.595	0.404	0.429	0.392
DQN	0.469	0.494	0.466	0.605	0.630	0.573	0.392	0.417	0.379
Multi-DQN	0.467	0.492	0.465	0.593	0.618	0.549	0.379	0.404	0.369
DLEF-SM	**0.466**	**0.491**	**0.464**	**0.581**	**0.606**	**0.528**	**0.367**	**0.392**	**0.335**

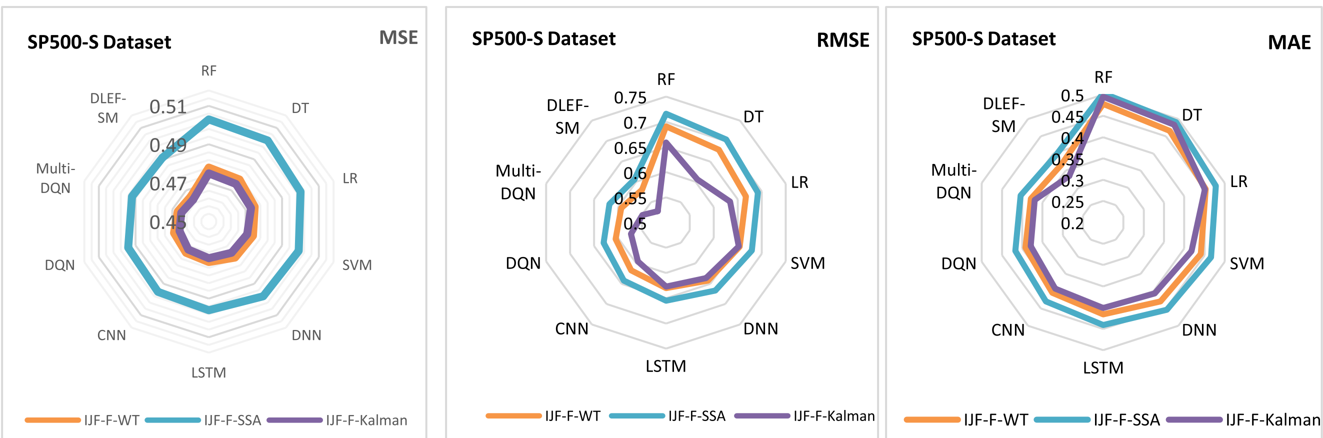


FIGURE 2. Error measures of the proposed framework with different preprocessing filtering techniques for SP500-S dataset.

model has the best performance for predicting the DAX stock index, as shown by its low RMSE and MAE relative to other models. After comparing the effects of three distinct pretreatment filtering strategies on model performance, it is clear that the IJF-F-WT framework provides the best results across all models and error metrics. This evidence points to wavelet-based filtering as a superior DAX dataset preparation option. In conclusion, the findings show that the proposed IJF-F-WT framework, which combines the DLEF-SM model with wavelet-based filtering, is an efficient method for predicting stock market indices like the DAX. Financial experts and investors might utilize this to their advantage when making choices based on expected stock market movements.

C. COMPARATIVE ANALYSIS WITH RESPECT TO QUALITY MEASURES

Table 6 showcases the outcomes of evaluating both the proposed and existing frameworks using the SP500-S dataset. Based on performance measures including F-measure and Sharpe ratio, the suggested framework is clearly superior than the state-of-the-art alternatives. The suggested framework DLEF-SM outperforms the best existing frameworks, by a wide margin, with an accuracy of 99.562%. The precision, recall, and F-measure also show similar trends, where the proposed framework outperforms the existing frameworks with significant margin. In terms of risk measures, the proposed framework achieves minimum drawdown (MDD) of 22.458%, which is the lowest among all the frameworks. Similar patterns can be seen with the coefficient of variation (COV) and the Sharpe ratio (SR), with the framework proposed achieving the best outcomes compared to the recent frameworks. As compared to other deep learning-based frameworks, the Multi-DQN and DLEF-SM perform very well. The proposed DLEF-SM system outperforms the state-of-the-art in terms of precision, volatility, and Sharpe ratio.

Table 7 presents the comparative analysis of proposed and existing frameworks for the SP500-L dataset based on various quality measures. Among the traditional methods,

TABLE 4. Results of proposed framework with different preprocessing filtering techniques for SP500-L dataset.

Forecasting frameworks	Error measures								
	MSE			RMSE			MAE		
	IJF-F-WT	IJF-F-SSA	IJF-F-Kalman	IJF-F-WT	IJF-F-SSA	IJF-F-Kalman	IJF-F-WT	IJF-F-SSA	IJF-F-Kalman
RF	0.490	0.515	0.488	0.704	0.729	0.751	0.490	0.515	0.488
DT	0.489	0.514	0.486	0.691	0.716	0.739	0.478	0.503	0.475
LR	0.488	0.513	0.485	0.679	0.704	0.726	0.466	0.491	0.463
SVM	0.486	0.511	0.484	0.667	0.692	0.714	0.453	0.478	0.451
DNN	0.485	0.510	0.482	0.654	0.679	0.702	0.441	0.466	0.438
LSTM	0.484	0.509	0.481	0.642	0.667	0.691	0.429	0.454	0.426
CNN	0.482	0.507	0.480	0.630	0.655	0.677	0.416	0.441	0.414
DQN	0.481	0.506	0.479	0.617	0.642	0.665	0.404	0.429	0.402
Multi-DQN	0.480	0.505	0.477	0.605	0.630	0.653	0.392	0.417	0.389
DLEF-SM	**0.478**	**0.503**	**0.476**	**0.593**	**0.618**	**0.640**	**0.379**	**0.404**	**0.377**

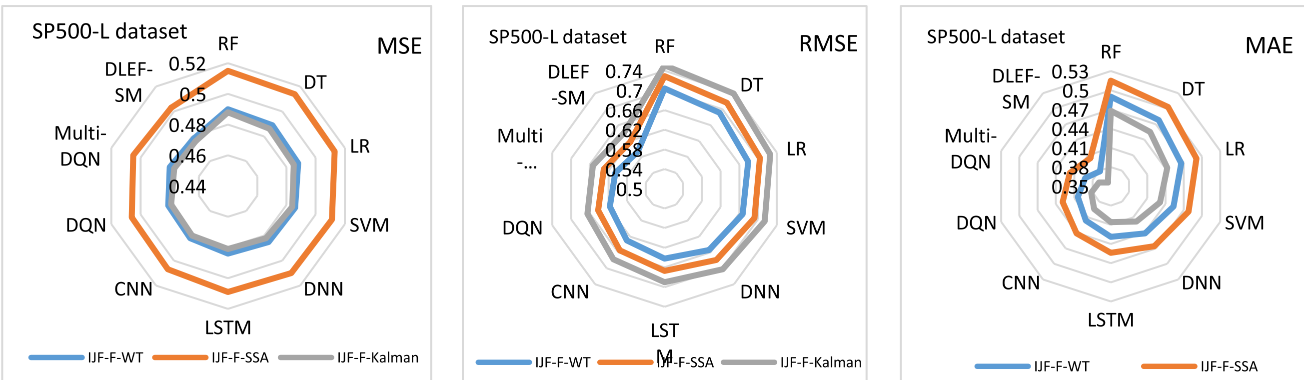


FIGURE 3. Error measures of the proposed framework with different preprocessing filtering techniques for SP500-L dataset.

TABLE 5. Results of proposed framework with different preprocessing filtering techniques for DAX dataset.

Forecasting frameworks	Error measures								
	MSE			RMSE			MAE		
	IJF-F-WT	IJF-F-SSA	IJF-F-Kalman	IJF-F-WT	IJF-F-SSA	IJF-F-Kalman	IJF-F-WT	IJF-F-SSA	IJF-F-Kalman
RF	0.502	0.527	0.500	0.776	0.741	0.713	0.502	0.527	0.47
DT	0.501	0.526	0.499	0.764	0.729	0.701	0.490	0.515	0.458
LR	0.500	0.525	0.497	0.751	0.716	0.689	0.478	0.503	0.445
SVM	0.499	0.524	0.496	0.739	0.704	0.676	0.466	0.491	0.433
DNN	0.497	0.522	0.495	0.727	0.692	0.664	0.453	0.478	0.421
LSTM	0.496	0.521	0.493	0.714	0.679	0.652	0.441	0.466	0.408
CNN	0.495	0.520	0.492	0.702	0.667	0.640	0.429	0.454	0.396
DQN	0.493	0.518	0.491	0.69	0.655	0.627	0.416	0.441	0.384
Multi-DQN	0.492	0.517	0.490	0.677	0.642	0.615	0.404	0.429	0.372
DLEF-SM	**0.491**	**0.516**	**0.488**	**0.665**	**0.630**	**0.603**	**0.392**	**0.417**	**0.359**

SVM achieved the highest accuracy of 83.485%, followed by DT with 81.235%. Among the deep learning methods, DLEF-SM achieved the highest accuracy of 98.235%, followed by Multi-DQN with 95.110% and CNN with 87.860%.

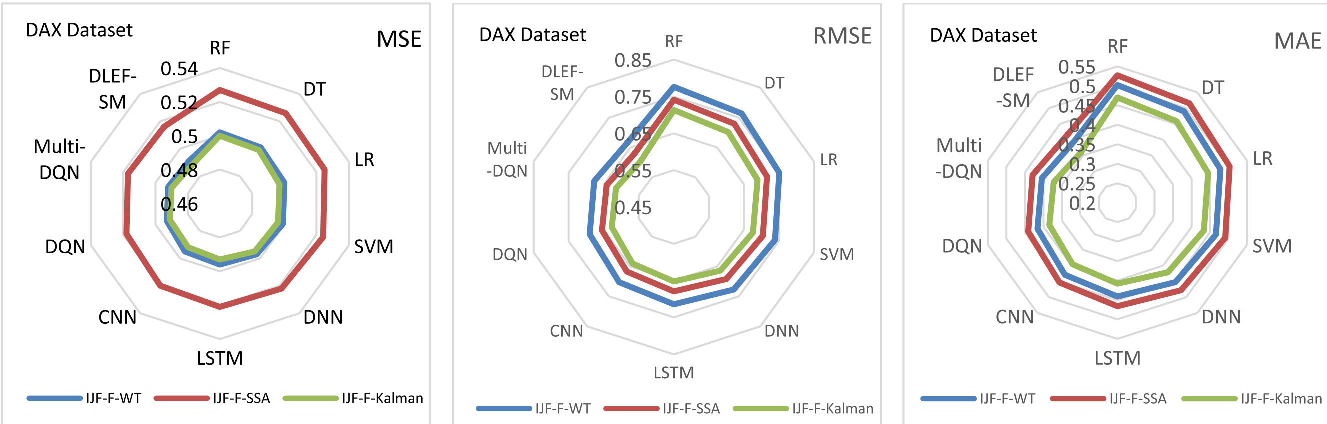


FIGURE 4. Error measures of the proposed framework with different preprocessing filtering techniques.

Compared to the results obtained for the SP500-S dataset, the accuracy scores for all methods decreased for the SP500-L dataset. Similarly, the MDD values increased for all methods, indicating higher risk in trading. Overall, the deep learning methods performed better than the traditional methods for both datasets. The proposed DLEF-SM method achieved the highest accuracy for the SP500-L dataset, outperforming all other methods. This indicates that DLEF-SM can be an effective method for long-term forecasting in financial trading. While the proposed technique has shown promise, its effectiveness may be affected by dataset details and the settings you decide to use.

Table 8 displays the results of a quality-measures-based comparison of both proposed and existing frameworks for the DAX dataset. All forecasting frameworks seem to perform better on this dataset than other two datasets, according to the findings. For this dataset the proposed DLEF-SM framework achieves the highest accuracy (98.825%), followed by Multi-DQN (96.910%).Comparing the results of the other traditional machine learning- based forecasting frameworks, SVM performs better than the other frameworks with an accuracy of 80.275%. The worst-performing frameworks are RF and DT, which have accuracies of 77.110% and 79.025%, respectively. In terms of MDD, COV, and SR, DLEF-SM outperforms all the other frameworks, indicating its superior performance in minimizing downside risk and maximizing returns. Similarly, Multi-DQN performs well on these measures, indicating its ability to provide stable returns with minimum downside risk. Overall, the proposed DLEF-SM framework outperforms all the other existing frameworks on the DAX dataset, followed by Multi-DQN. However, CNN and LSTM perform better than the other existing frameworks.

Figure 5 shows the accuracy measure based comparison of various forecasting frameworks with respect to three different datasets. Similarly, Figure 6 clearly illustrates the performance analysis of the proposed framework DLEF-SM in terms of F-measure along with existing frameworks with three different datasets. Figure 7 shows the computation time for the various forecasting frameworks. Our DLEF-SM framework excels in terms of computation time with 2.235Sec.

D. ABLATION STUDY

Our improved black widow optimization (IBWO) algorithm significantly enhances the performance of various forecasting frameworks across different datasets, as shown in Table 9. For instance, the accuracy of Random Forest for the SP500-S dataset increased from 75.777 to 79.437, and the accuracy of Multi-DQN for the DAX dataset improved from 93.250 to 96.910. These results indicate the IBWO algorithm's effectiveness in optimizing features and significantly boosting the performance of stock market forecasting models. As shown in Table 10, the statistical tests provide robust evidence that the proposed DLEF-SM framework performs significantly better than the existing Multi-DQN framework across a range of quality measures in stock market forecasting. These results underscore the efficacy of DLEF-SM in improving accuracy, precision, recall, F-measure, MDD, COV, and SR, thereby shown its potential superiority in predictive performance and risk management in financial markets. Residual analysis is crucial in evaluating the performance of the proposed DLEF-SM framework for price prediction. Figure 8 illustrates the comparison between predicted and expected prices, with residuals plotted along the y-axis. The plot enables assessment of how well the model predictions align with actual prices.

The K-fold cross-validation results in Table 11 demonstrate the robust performance of the proposed DLEF-SM framework for stock market forecasting across multiple metrics and datasets. Each row represents a different fold of the cross-validation process, ensuring that the model's performance is evaluated comprehensively. Across all datasets (SP500-S, SP500-L, and DAX), the framework consistently achieves high accuracy, precision, and recall percentages, indicating its ability to generalize well to unseen data. The progressive improvement in metrics from the first to the tenth fold suggests that the model effectively learns from the data without overfitting. These results underscore the reliability

TABLE 6. An evaluation of the proposed and existing frameworks for the SP500-S dataset.

Forecasting frameworks	Quality measures (%)						
	Accuracy	Precision	Recall	F-measure	MDD	COV	SR
RF	79.437	78.112	77.931	78.021	78.026	41.583	79.020
DT	82.562	80.237	80.056	82.146	80.151	39.458	81.145
LR	83.687	82.362	82.181	84.271	82.276	37.333	83.270
SVM	88.812	84.487	84.306	87.396	84.401	35.208	85.395
DNN	87.937	86.612	86.431	91.521	86.526	33.083	87.520
LSTM	90.062	88.737	88.556	90.646	88.651	30.958	89.645
CNN	91.187	90.862	90.681	91.771	90.776	28.833	91.770
DQN	94.312	92.987	92.806	90.896	92.901	26.708	93.895
Multi-DQN	97.437	95.112	94.931	98.021	95.026	24.583	96.020
DLEF-SM	**99.562**	**97.237**	**97.056**	**99.146**	**97.151**	**22.458**	**98.145**

TABLE 7. Comparative analysis of proposed and existing frameworks for SP500-L dataset.

Forecasting frameworks	Quality measures (%)							
	Accuracy	Precision	Recall	F-measure	MDD	COV	SR	Computation time (s)
RF	77.110	77.002	77.003	77.002	77.921	37.360	76.923	4.562
DT	81.235	79.127	79.128	79.127	80.046	35.235	79.048	6.124
LR	79.360	81.252	81.253	81.252	82.171	33.110	81.173	5.368
SVM	83.485	83.377	83.378	83.377	84.296	30.985	83.298	4.652
DNN	86.610	85.502	85.503	85.502	86.421	28.860	85.423	7.125
LSTM	84.735	87.627	87.628	87.627	88.546	26.735	87.548	7.236
CNN	87.860	89.752	89.753	89.752	90.671	24.610	89.673	6.487
DQN	91.985	91.877	91.878	91.877	92.796	22.485	91.798	5.623
Multi-DQN	95.110	94.002	94.003	94.002	94.921	20.360	93.923	4.562
DLEF-SM	**98.235**	**96.127**	**96.128**	**96.127**	**97.046**	**18.235**	**96.048**	**2.235**

TABLE 8. Comparative analysis of proposed and existing frameworks for DAX dataset.

Forecasting frameworks	Quality measures (%)						
	Accuracy	Precision	Recall	F-measure	MDD	COV	SR
RF	77.910	77.989	78.523	79.255	78.749	38.473	78.023
DT	79.025	80.114	80.648	80.380	80.874	36.348	80.148
LR	81.150	82.239	82.773	84.505	82.999	34.223	82.273
SVM	80.275	84.364	84.898	84.631	85.124	32.098	84.398
DNN	87.401	86.489	87.023	86.755	87.249	29.973	86.523
LSTM	87.525	88.614	89.148	89.880	89.374	27.848	88.648
CNN	90.650	90.739	91.273	91.005	91.499	25.723	90.773
DQN	92.775	92.864	93.398	95.130	93.624	23.598	92.898
Multi-DQN	96.910	94.989	95.523	96.255	95.749	21.473	95.023
DLEF-SM	**98.825**	**97.114**	**97.648**	**97.382**	**97.874**	**19.348**	**97.148**

and stability of the DLEF-SM framework in predicting stock market trends, validating its potential utility in practical applications where accurate and consistent forecasting is essential for informed decision-making.

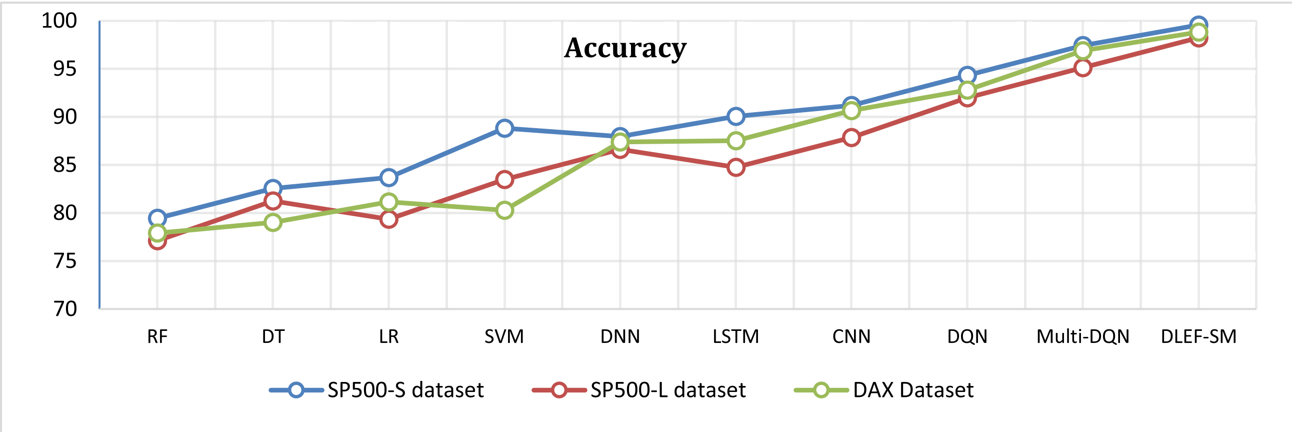


FIGURE 5. Performance comparison (accuracy) of proposed and existing forecasting frameworks with SP500-S, SP500-Land DAX datasets.

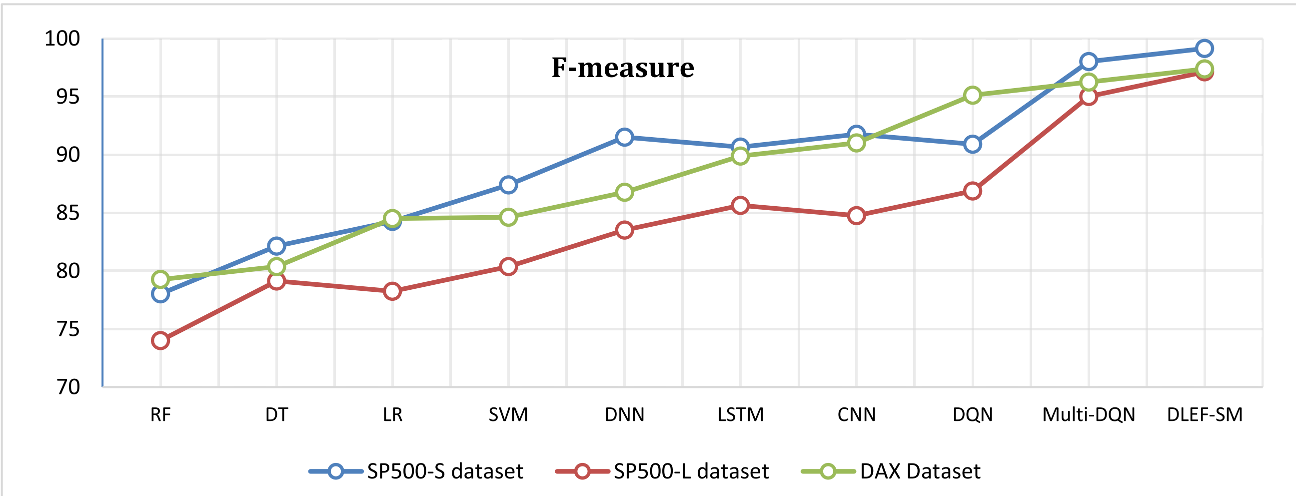


FIGURE 6. Performance comparison (F-Measure) of proposed and existing forecasting frameworks with SP500-S, SP500-Land DAX datasets.

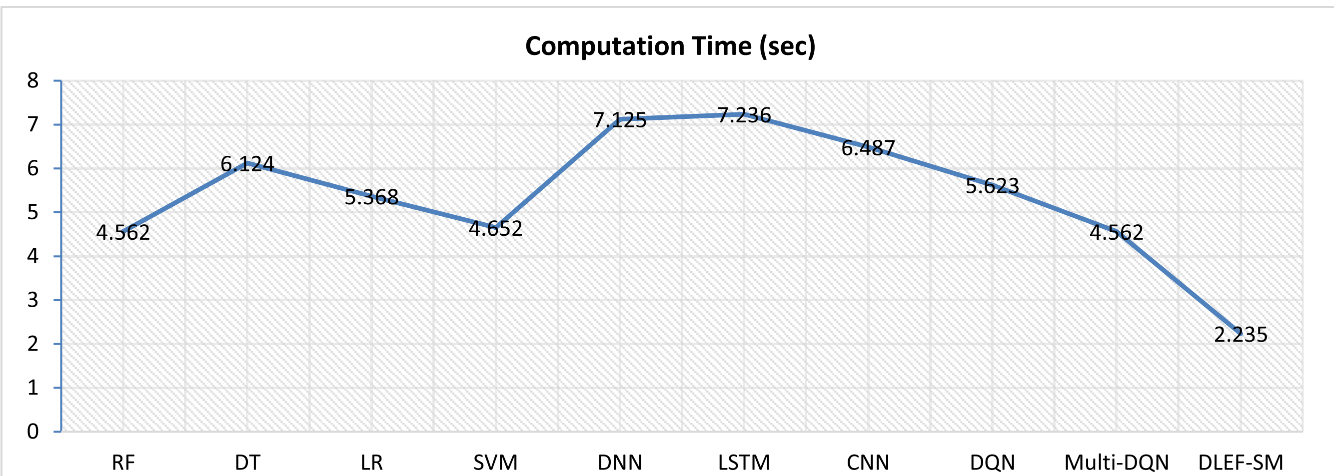


FIGURE 7. Computation time of forecasting frameworks for SP500-L dataset.

E. DISCUSSIONS

From the simulation results, we observed the following improvement of the proposed framework over the existing frameworks.

1. For the SP500-S dataset, the DLEF-SM framework achieved an accuracy of 99.562%, which was 2.125%, 5.25%, 8.375% and 9.5% higher than the second-best performing framework, Multi-DQN (97.437%), and the

TABLE 9. Accuracy impact of feature optimization for forecasting frameworks with different datasets.

Forecasting frameworks	Without IBWO algorithm			With IBWO algorithm		
	SP500-S	SP500-L	DAX	SP500-S	SP500-L	DAX
RF	75.777	73.450	74.250	79.437	77.110	77.910
DT	78.902	77.575	75.365	82.562	81.235	79.025
LR	80.027	75.700	77.490	83.687	79.360	81.150
SVM	85.152	79.825	76.615	88.812	83.485	80.275
DNN	84.277	82.950	83.741	87.937	86.610	87.401
LSTM	86.402	81.075	83.865	90.062	84.735	87.525
CNN	87.527	84.200	86.990	91.187	87.860	90.650
DQN	90.652	88.325	89.115	94.312	91.985	92.775
Multi-DQN	93.777	91.450	93.250	97.437	95.110	96.910
DLEF-SM	95.902	94.575	95.165	**99.562**	**98.235**	**98.825**

TABLE 10. Statistical tests proposed DLEF-SM and existing Multi-DQN frameworks for stock market forecasting.

Quality measure (%)	t-Statistic			P-Value		
	SP500-S	SP500-L	DAX	SP500-S	SP500-L	DAX
Accuracy	4.9200	7.6200	9.6200	0.0002	0.0001	0.0001
Precision	4.3800	6.9800	8.9800	0.0004	0.0002	0.0002
Recall	3.7200	6.1100	8.1100	0.0015	0.0003	0.0003
F-measure	5.2100	7.9200	9.9200	0.0001	0.0001	0.0001
MDD	2.1500	5.3600	6.3600	0.0357	0.0005	0.0005
COV	3.6200	4.7200	5.7200	0.0008	0.0007	0.0007
SR	4.0500	6.2800	7.2800	0.0007	0.0002	0.0002

TABLE 11. K-fold cross validation of proposed DLEF-SM framework for stock market forecasting.

k-fold	SP500-S	SP500-L	DAX	SP500-S	SP500-L	DAX	SP500-S	SP500-L	DAX
	Accuracy (%)			Precision (%)			Recall (%)		
1	89.950	89.257	88.846	89.139	88.467	87.978	88.483	87.180	87.587
2	91.006	90.399	89.988	90.195	89.609	89.120	89.539	88.322	88.729
3	92.063	91.541	91.130	91.252	90.751	90.262	90.596	89.464	89.871
4	93.119	92.683	92.272	92.308	91.893	91.404	91.652	90.606	91.013
5	94.175	93.825	93.414	93.364	93.035	92.546	92.708	91.748	92.155
6	95.231	94.967	94.556	94.420	94.177	93.688	93.764	92.890	93.297
7	96.287	96.109	95.698	95.476	95.319	94.830	94.820	94.032	94.439
8	97.344	97.251	96.840	96.533	96.461	95.972	95.877	95.174	95.581
9	98.400	98.393	97.982	97.589	97.603	97.114	96.933	96.316	96.723
10	99.456	99.535	99.124	98.645	98.745	98.256	97.989	97.458	97.865

existing frameworks DQN, CNN, and LSTM respectively. Moreover, the DLEF-SM framework achieved the highest precision, recall, and F-measure scores of 97.114, 97.648 and 97.382respectively, which were significantly higher than those of other existing frameworks. Additionally, DLEF-SM exhibited the highest Maximum Drawdown (MDD) and Coefficient of Variation (COV) scores, reflecting its stability and consistency in performance. Moreover, it achieved the highest Sharpe Ratio (SR) score, suggesting favorable risk-adjusted returns.

2. For the SP500-L dataset, the DLEF-SM framework achieved an accuracy of 98.235%, which was 1.915%, 6.05%, and 10.370% higher than the second-best performing frameworks, Multi-DQN (96.910%), DQN (91.985%) and CNN (87.860). The DLEF-SM framework also achieved the highest precision, recall, and F-measure scores of 96.127, 96.128, and 96.127,respectively, which were significantly higher than those of other existing frameworks. The results reveals that in the SP500-L dataset, the DLEF-SM framework demonstrated notable performance metrics, including

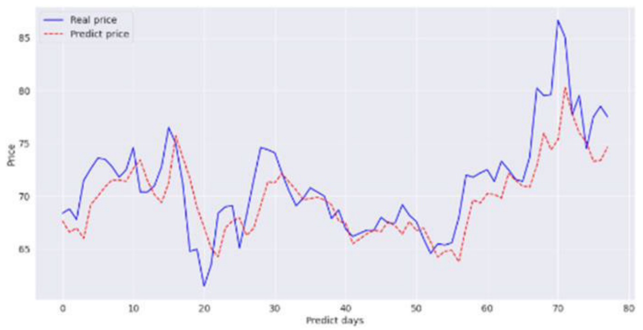


FIGURE 8. Residual analysis of proposed DLEF-SM framework with predicted and expected price.

the highest MDD and COV scores, reaching 97.046% and 18.235% respectively, along with the highest SR score of 96.048%. A high MDD score indicates that the framework experienced minimal losses relative to its peak value during the evaluation period, showcasing its ability to mitigate downside risk effectively. The low COV score suggests that the framework achieved consistent returns relative to its mean performance, indicating stability and reliability in its investment outcomes. Additionally, the elevated SR score underscores the framework's strong risk-adjusted returns, implying that it generated significant returns considering the level of risk undertaken. These findings collectively suggest that the DLEF-SM framework exhibited robust performance characteristics in the SP500-L dataset, making it potentially attractive for investment strategies requiring both strong returns and prudent risk management.

3. For the DAX dataset, the DLEF-SM framework achieved an accuracy of 98.825%, which was 1.915%, 6.05%, and 8.175% higher than the second-best performing frameworks, Multi-DQN (96.910%), DQN (92.775%), and CNN (90.650%). The DLEF-SM framework also achieved the highest precision, recall, and F-measure scores of 97.114,97.648 and 97.382 respectively, which were significantly higher than those of other existing frameworks. The results indicates that the DLEF-SM framework attained the highest MDD and COV scores, standing at 97.874% and 19.348% respectively, alongside the highest SR score of 97.148%. A high MDD score signifies minimal losses experienced by the framework during the evaluation period, reflecting its adeptness in mitigating downside risk and preserving capital. Moreover, the low COV score suggests that the framework achieved consistent returns relative to its mean performance, showcasing stability and predictability. The elevated SR score underscores the framework's exceptional risk-adjusted returns, indicating its ability to generate significant returns relative to the level of risk taken. These findings collectively suggest that the DLEF-SM framework not only demonstrated strong absolute performance but also excelled in managing risk, making it potentially appealing for investment strategies necessitating both high returns and prudent risk management.
4. The computation time comparison across various forecasting frameworks applied to the SP500-L dataset reveals notable differences in efficiency. Simpler models such as Linear Regression and Random Forest demonstrate relatively low computation times, showcasing their scalability and efficiency for this dataset. Support Vector Machine and Multi-Deep Q-Network also exhibit moderate computation times, indicating their suitability for this task. However, more complex models like Deep Neural Networks, LSTM, and Convolutional Neural Networks require significantly more computation time, highlighting the trade-off between model complexity and computational efficiency. Notably, the DLEF-SM framework stands out with the shortest computation time, suggesting its potential for efficient forecasting tasks on this dataset. Overall, the choice of forecasting framework should consider both prediction performance and computational efficiency, particularly for large datasets or real-time applications.

Overall, the statistical comparative analysis clearly indicates that the DLEF-SM framework is the most effective and accurate framework for stock price prediction among all other proposed and existing frameworks

V. CONCLUSION

We have introduced a novel framework, DLEF-SM, for stock market forecasting that integrates deep learning and efficient feature optimization techniques. The framework leverages an improved jellyfish-induced filtering technique for preprocessing, pre-trained CNN architectures for feature extraction, and an enhanced black widow optimization algorithm for feature selection. Our approach utilizes a hybrid DRL-ANN model to achieve high accuracy in stock market forecasting [36], [37]. Validation was conducted using standard measures such as S&P500-S, S&P500-L, and the DAX market, demonstrating that our framework outperforms existing state-of-the-art approaches in accuracy, precision, recall, F-measure, MDD, COV, and SR. Across both datasets, the experimental results consistently show the superiority of DLEF-SM. The DLEF-SM framework represents an advanced system for stock market forecasting, effectively incorporating deep learning, efficient feature optimization, and deep reinforcement learning techniques [38], [39]. This allows the model to handle large datasets, extract meaningful features, and mitigate under-fitting by reducing data dimensionality. Moreover, DLEF-SM demonstrates adaptability to market dynamics, enabling informed decision-making based on historical data, which is valuable for financial analysts and traders. However, it's important to acknowledge the limitations of our study. Our work lacks validation on extremely dense and real-time datasets, which are crucial in practical financial forecasting scenarios. Future research could explore the performance of different deep learning architectures in handling such data, emphasizing their ability to capture temporal dependencies in financial markets. Addressing these limitations would further enhance the applicability and robustness of DLEF-SM in real-world financial forecasting environments.

DECLARATION OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

[1] P.-C. Chang and C.-Y. Fan, ''A hybrid system integrating a wavelet and TSK fuzzy rules for stock price forecasting,'' *IEEE Trans. Syst. Man, Cybern., Part C (Appl. Rev.)*, vol. 38, no. 6, pp. 802–815, Nov. 2008.

[2] L. Yu, H. Chen, S. Wang, and K. Keung Lai, ''Evolving least squares support vector machines for stock market trend mining,'' *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 87–102, Feb. 2009.

[3] C. Slamka, B. Skiera, and M. Spann, ''Prediction market performance and market liquidity: A comparison of automated market makers,'' *IEEE Trans. Eng. Manag.*, vol. 60, no. 1, pp. 169–185, Feb. 2013.

[4] J. Alexandre Macedo, L. Theodoro Oliveira Camargo, H. Cesar Brandao de Oliveira, L. Eduardo da Silva, and R. Menezes Salgado, ''An intelligent decision support system to investment in the stock market,'' *IEEE Latin Amer. Trans.*, vol. 11, no. 2, pp. 812–819, Mar. 2013.

[5] S.-M. Chen and S.-W. Chen, ''Fuzzy forecasting based on two-factors second-order fuzzy-trend logical relationship groups and the probabilities of trends of fuzzy logical relationships,'' *IEEE Trans. Cybern.*, vol. 45, no. 3, pp. 391–403, Mar. 2015.

[6] A. Tenyakov, R. Mamon, and M. Davison, ''Filtering of a discrete-time HMM-driven multivariate ornstein-uhlenbeck model with application to forecasting market liquidity regimes,'' *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 6, pp. 994–1005, Sep. 2016.

[7] J.-M. Le Caillec, A. Itani, D. Guériot, and Y. Rakotondratsimba, ''Stock picking by probability–possibility approaches,'' *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 2, pp. 333–349, Apr. 2017.

[8] G. G. Creamer, ''Network structure and market risk in the European equity market,'' *IEEE Syst. J.*, vol. 12, no. 2, pp. 1090–1098, Jun. 2018.

[9] J.-S. Chou and T.-K. Nguyen, ''Forward forecast of stock price using sliding-window metaheuristic-optimized machine-learning regression,'' *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3132–3142, Jul. 2018.

[10] Y. Guo, S. Han, C. Shen, Y. Li, X. Yin, and Y. Bai, ''An adaptive SVR for high-frequency stock price forecasting,'' *IEEE Access*, vol. 6, pp. 11397–11404, 2018.

[11] Y. Chen, K. He, and G. K. F. Tso, ''Forecasting crude oil prices: A deep learning based model,'' *Proc. Comput. Sci.*, vol. 122, pp. 300–307, Sep. 2017.

[12] O. B. Sezer, M. Ozbayoglu, and E. Dogdu, ''A deep neural-network based stock trading system based on evolutionary optimized technical analysis parameters,'' *Proc. Comput. Sci.*, vol. 114, pp. 473–480, Sep. 2017.

[13] E. Hoseinzade and S. Haratizadeh, ''CNNpred: CNN-based stock market prediction using a diverse set of variables,'' *Expert Syst. Appl.*, vol. 129, pp. 273–285, Sep. 2019.

[14] W. Chen, C. K. Yeo, C. T. Lau, and B. S. Lee, ''Leveraging social media news to predict stock index movement using RNN-boost,'' *Data Knowl. Eng.*, vol. 118, pp. 14–24, Nov. 2018.

[15] D. Pradeepkumar and V. Ravi, ''Forecasting financial time series volatility using particle swarm optimization trained quantile regression neural network,'' *Appl. Soft Comput.*, vol. 58, pp. 35–52, Sep. 2017.

[16] E. Chong, C. Han, and F. C. Park, ''Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies,'' *Expert Syst. Appl.*, vol. 83, pp. 187–205, Oct. 2017.

[17] F. Milani, ''Learning about the interdependence between the macro economy and the stock market,'' Dept. of Economics Working Papers, Univ. California, Rep., 2008.

[18] W. Zhang, S. Zhang, S. Zhang, D. Yu, and N. Huang, ''A multi-factor and high-order stock forecast model based on type-2 FTS using cuckoo search and self-adaptive harmony search,'' *Neurocomputing*, vol. 240, pp. 13–24, May 2017.

[19] J. Wang and J. Wang, ''Forecasting stochastic neural network based on financial empirical mode decomposition,'' *Neural Netw.*, vol. 90, pp. 8–20, Jun. 2017.

[20] D. Cabrera-Paniagua, C. Cubillos, R. Vicari, and E. Urra, ''Decision-making system for stock exchange market using artificial emotions,'' *Expert Syst. Appl.*, vol. 42, no. 20, pp. 7070–7083, Nov. 2015.

[21] Y. Li, P. Ni, and V. Chang, ''Application of deep reinforcement learning in stock trading strategies and stock forecasting,'' *Computing*, vol. 102, no. 6, pp. 1305–1322, Jun. 2020.

[22] S. Ding, T. Cui, X. Xiong, and R. Bai, ''Forecasting stock market return with nonlinearity: A genetic programming approach,'' *J. Ambient Intell. Humanized Comput.*, vol. 11, no. 11, pp. 4927–4939, Nov. 2020.

[23] S. W. Lee and H. Y. Kim, ''Stock market forecasting with super-high dimensional time-series data using ConvLSTM, trend sampling, and specialized data augmentation,'' *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113704.

[24] Š. Lyócsa and D. Stasek, ''Improving stock market volatility forecasts with complete subset linear and quantile HAR models,'' *Expert Syst. Appl.*, vol. 183, Nov. 2021, Art. no. 115416.

[25] Y. Lin, Y. Yan, J. Xu, Y. Liao, and F. Ma, ''Forecasting stock index price using the CEEMDAN-LSTM model,'' *North Amer. J. Econ. Finance*, vol. 57, Jul. 2021, Art. no. 101421.

[26] S. K. Chandar, ''Hybrid models for intraday stock price forecasting based on artificial neural networks and metaheuristic algorithms,'' *Pattern Recognit. Lett.*, vol. 147, pp. 124–133, Jul. 2021.

[27] A. U. Haq, A. Zeb, Z. Lei, and D. Zhang, ''Forecasting daily stock trend using multi-filter feature selection and deep learning,'' *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114444.

[28] D. O. Oyewola, A. Ibrahim, J. A. Kwanamu, and E. G. Dada, ''A new auditory algorithm in stock market prediction on oil and gas sector in Nigerian stock exchange,'' *Soft Comput. Lett.*, vol. 3, Dec. 2021, Art. no. 100013.

[29] M. Shahvaroughi Farahani and S. H. Razavi Hajiagha, ''Forecasting stock price using integrated artificial neural network and metaheuristic algorithms compared to time series models,'' *Soft Comput.*, vol. 25, no. 13, pp. 8483–8513, Jul. 2021.

[30] D. K. Sharma, H. S. Hota, and A. R. Rababaah, ''Forecasting U.S. stock price using hybrid of wavelet transforms and adaptive neuro fuzzy inference system,'' *Int. J. Syst. Assurance Eng. Manage.*, vol. 15, no. 2, pp. 591–608, Feb. 2024.

[31] S. Carta, A. Ferreira, A. S. Podda, D. Reforgiato Recupero, and A. Sanna, ''Multi-DQN: An ensemble of deep Q-learning agents for stock market forecasting,'' *Expert Syst. Appl.*, vol. 164, Feb. 2021, Art. no. 113820.

[32] N. Deepika and M. Nirupama Bhat, ''An efficient stock market prediction method based on Kalman filter,'' *J. Inst. Engineers (India), Ser. B*, vol. 102, pp. 629–644, Apr. 2021.

[33] Q. Tang, T. Fan, R. Shi, J. Huang, and Y. Ma, ''Prediction of financial time series using LSTM and data denoising methods,'' 2021, *arXiv:2103.03505*.

[34] R. Tr, U. K. Lilhore, S. Simaiya, A. Kaur, and M. Hamdi, ''Predictive analysis of heart diseases with machine learning approaches,'' *Malaysian J. Comput. Sci.*, vol. 1, pp. 132–148, Mar. 2022.

[35] P. Dhiman, A. Kaur, C. Iwendi, and S. K. Mohan, ''A scientometric analysis of deep learning approaches for detecting fake news,'' *Electronics*, vol. 12, no. 4, p. 948, Feb. 2023.

[36] D. S. Dhaliwal, P. S. Sandhu, and S. N. Panda, ''Enhanced K-nearest neighbor algorithm,'' *World Acad. Sci., Eng. Technol.*, vol. 73, pp. 681–685, Aug. 2011.

[37] U. K. Lilhore, M. Poongodi, A. Kaur, S. Simaiya, A. D. Algarni, H. Elmannai, and M. Hamdi, ''Hybrid model for detection of cervical cancer using causal analysis and machine learning techniques,'' *Comput. Math. Methods Med.*, vol. 2022, no. 1, 2022, Art. no. 4688327.

[38] K. Verma, S. Bhardwaj, R. Arya, M. S. U. Islam, M. Bhushan, A. Kumar, and P. Samant, ''Latest tools for data mining and machine learning,'' *Int. J. Innov. Technol. Exploring Eng.*, vol. 8, no. 9, pp. 18–23, 2019.

[39] P. K. Shukla, J. K. Sandhu, A. Ahirwar, D. Ghai, P. Maheshwary, and P. K. Shukla, ''Multiobjective genetic algorithm and convolutional neural network based COVID-19 identification in chest X-ray images,'' *Math. Problems Eng.*, vol. 2021, pp. 1–9, Feb. 2021.

[40] S. S. Pal and S. Kar, ''Time series forecasting for stock market prediction through data discretization by fuzzistics and rule generation by rough set theory,'' *Math. Comput. Simul.*, vol. 162, pp. 18–30, Aug. 2019.

[41] S. Guo, L. Yu, X. Li, and S. Kar, ''Fuzzy multi-period portfolio selection with different investment horizons,'' *Eur. J. Oper. Res.*, vol. 254, no. 3, pp. 1026–1035, Nov. 2016.

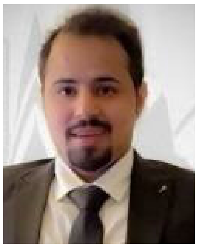
[42] M. B. Kar, S. Kar, S. Guo, X. Li, and S. Majumder, "A new bi-objective fuzzy portfolio selection model and its solution through evolutionary algorithms," *Soft Comput.*, vol. 23, no. 12, pp. 4367–4381, Jun. 2019.
[43] S. S. Pal and S. Kar, "Time series forecasting using fuzzy transformation and neural network with back propagation learning," *J. Intell. Fuzzy Syst.*, vol. 33, no. 1, pp. 467–477, Jun. 2017.

FATHE JERIBI received the M.S. degree in computer science and information technology from Sacred Heart University, Connecticut, USA, and the Ph.D. degree in information technology from Towson University, MD, USA. He is an Associate Professor with the College of Computer Science and Information Technology, Jazan University, Saudi Arabia. His areas of interests include machine learning, software engineering, computer networks, SDN, wireless ad hoc networks, the IoT, and distributed computing.

R. JOHN MARTIN received the Ph.D. degree in computer science from Bharathiar University, India, focused on an innovative machine learning model for epileptic seizure detection using EEG signals. He is a Distinguished Educator in computer science and holds an expertise in machine learning, healthcare AI, and signal processing, spanning a 27 years of career in higher education, research, and educational leadership. Currently, he is with Jazan University, Saudi Arabia. He is known for his pedagogical talent and administrative leadership. His impactful works feature in prestigious journals of IEEE and ACM. Notably, he holds international patents and copyrights, showcasing ground-breaking healthcare AI innovations.

RUCHI MITTAL is a professional with over two decades of work experience in teaching, training, research, and academic administration. Her teaching expertise comprises of DBMS, machine learning, social networking analysis, and data mining. She is an active Ph.D. advisor. She has published over 40 articles in Scopus/SCI-indexed journals, book series, and conference proceedings. She received the Chitkara University Research Excellence Award in author with highest H-index category, in 2020 (Computer Applications Discipline). Currently, she reviews for a number of leading international journals, such as *Internet Research* (Emerald), *World Review of Science, Technology and Sustainable Development* (Inderscience), *International Journal of Computational Science and Engineering* (Inderscience), *International Journal of Intelligent Systems Technologies and Applications* (Inderscience), and a number of IEEE and Springer conferences. She serves on the TPC for a number of conferences.

HASSAN JARI received the B.Sc. degree in computer information technology from Purdue University, Indianpolis, USA, the M.Sc. degree in computer, network and information security from DePaul University, Chicago, USA, and the Ph.D. degree in computer science from Newcastle University, Newcastle, U.K. He is an Assistant Professor with the College of Engineering and Computer Science, Jazan University, Saudi Arabia. His areas of interests include cloud computing, the IoT, computer networks, artificial intelligence, IT security, and performance engineering.



ABDULRAHMAN HASSAN ALHAZMI received the M.S. degree in software engineering from Gannon University, USA, and the Ph.D. degree in software and application security from La Trobe University, Melbourne, Australia. He is an Assistant Professor with the College of Engineering and Computer Science, Jazan University, Saudi Arabia. His areas of interests include machine learning, software engineering, security and privacy of applications, serious games for cyber security, human–computer interaction, and computer networks.

VARUN MALIK (Member, IEEE) received the Ph.D. degree in computer science and engineering from I. K. Gujral Punjab Technical University, Punjab, India. He is holding more than ten Indian patents/copyrights. His current research interests include blockchain, artificial intelligence, cloud computing, cyber security, the Internet of Things, data mining and warehousing, and method engineering. He is serving as a reviewer for many international journals.

S. L. SWAPNA received the master's and M.Phil. degrees in computer science from Manonmaniam Sundaranar University, India, and the Ph.D. degree in computer science from Bharathiar University, Coimbatore, India. She is an Academician and a Technical Trainer. She is currently a professional IT Trainer with Applexus Technologies, Technopark, Thiruvananthapuram. Previously, she was a Faculty Member with the School of Engineering and Computer Science, Jazan University, Saudi Arabia. Her research interests include big data, machine learning, predictive data analytics, and privacy preserved data security.

S. B. GOYAL (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from Banasthali University, Rajasthan, India, in 2012. He is currently the Director of the Faculty of Information Technology, City University Malaysia. He was with many institutions in many different academic and administrative positions. He is holding more than 20 years of work experience at national and international levels. He has peerless inquisitiveness and enthusiasm to get abreast with the latest development in the IT field. He has good command over industry revolution (IR) 4.0 technologies, such as big data, data science, artificial intelligence and blockchain, and cloud computing. He is the first one to introduce IR 4.0, including blockchain technology in the academic curriculum in Malaysian universities. He has participated in many panel discussions on IR 4.0 technologies at academia and industry platforms. He has contributed in many Scopus/SCI journals/conferences. He is holding more than ten international patents/copyrights from Australia, Germany, and India. His current research interests include blockchain, artificial intelligence, cloud computing, cyber security, the Internet of Things, data mining and warehousing, and method engineering.

MANOJ KUMAR (Member, IEEE) received the B.Tech. degree in computer science from Kurukshetra University, the M.Sc. degree in information security and forensics from ITB, Dublin, the M.Tech. degree from ITM University, and the Ph.D. degree in computer science from The NorthCap University, Gurugram. He was associated as an Assistant Professor with the Computer Science Department, Amity University, Noida, Uttar Pradesh. He is currently an Assistant Professor (SG) (SoCS) with the University of Petroleum and Energy Studies, Dehradun. He has over nine years of experience in research and academics. He has published over 25 publications in reputed journals and conferences. He is a member of various professional bodies, including ACM and ISTE. He was a reviewer of many reputed journals.

SHUBHRANSHU VIKRAM SINGH (Senior Member, IEEE) received the B.Tech. degree in electrical engineering from Uttar Pradesh Technical University, and the M.Tech. degree from SVN University, Sagar. He is with Amity University, Greater Noida Campus, as an Associate Professor. He has collaborated with eminent professors across the world from top QS-ranked universities. He has published over 40 articles in peer-reviewed journals and conferences of international repute, such as Elsevier, Springer, IEEE, and Wiley. Apart from that, several book chapters. He has several Indian patents and copyrights under his credit. He has given several talks at various events and symposiums. He has also served on the publication team of several conferences of internationally reputed journals and a member of various society. His current research interests include the IoT, AI, and power management. He is also working on solar power management. He is the Organizer of several IEEE conferences, such as ICIPTM2022, ICIEM2022, ICCAKM2022, ICTACS, IC3I, and UPCON2023.