



# Comprehensive Analysis of Stock Price Dynamics Using Ensemble Machine Learning Models for Enhanced Prediction Accuracy

Vinamra Nayak<sup>1,\*</sup>

<sup>1</sup> Associate Professor, GGITS, Jabalpur, M.P., India

Email: [vinamranayak@ggits.org](mailto:vinamranayak@ggits.org)

Received: June 10, 2024 Revised: September 25, 2024 Accepted: December 10, 2024 ★ Corresponding author

## ABSTRACT

Stock price prediction is an important component of financial analysis because the results influence economic growth and investment. This work aims to develop an ensemble supervised-learning technique consisting mainly of Principal Component Analysis (PCA), Particle Swarm Optimization (PSO), and a Support Vector Machine (SVM) to achieve better prediction. Through PCA, the dimensions of large stock datasets are compressed without compromising crucial features. The problem of parameter selection for nonlinear datasets is handled using the bio-inspired PSO technique to optimize the SVM hyperparameters. As the core predictor, the SVM employs a Radial Basis Function to provide substantial regression capacity for sophisticated financial datasets. The ensemble framework was evaluated using actual stock-price data divided into training and testing sets. The results indicate that the proposed approach is more accurate than conventional approaches, with an accuracy rate of 95.5%, and achieves improved RMSE and MAE values. In particular, integrating PCA for feature reduction and PSO for parameter tuning with SVM regression provides a notable improvement in stock-price forecasting. The proposed methodology can be scaled for financial analytics because it addresses noisy, nonlinear, high-dimensional data.

**Keywords:** PCA ▪ PSO ▪ SVM ▪ Stock Price Prediction ▪ Financial Data Analysis ▪ Regression Models ▪ Machine Learning in Finance ▪ Stock Market Trends ▪ Dimensionality Reduction

## 1. INTRODUCTION

Predicting stock prices has remained a widely studied problem in investment and computer science. Investors, analysts, and policy makers need efficient tools through which informed decisions—from trading strategies to risk management and market analysis—can be made. However, the volatility and nonlinearity of financial markets make stock-price prediction inherently difficult [1]. The factors affecting price levels are numerous, ranging from macroeconomic parameters to firm-specific information, market sentiment, and geopolitical events.

One of the principal challenges in stock-price prediction is capturing the complex interactions among these factors while working with data that are normally noisy and high-dimensional [2]. Traditional statistical techniques for time-series analysis can be useful for some tasks but cannot fully capture such complications. For instance, ARIMA and simple linear-regression models assume stationarity and linearity, which may not hold in financial markets [3]. These limitations have opened the way for advanced methods based on machine learning (ML) and computational intelligence.

Artificial intelligence has changed how stock-price prediction is approached. Algorithms designed to learn from data

and discover patterns can provide more accurate and timely results [4]. In contrast to classical statistical models, ML models do not necessarily require specific assumptions about the distribution of the underlying sample data, which provides flexibility in nonlinear financial markets [5].

The variety of ML approaches makes it possible to apply different methods to stock-price prediction. Neural networks, decision trees, and SVMs are supervised-learning models that use historical information to learn relationships with future price movements [6]. Useful unsupervised methods include clustering and dimensionality-reduction techniques. Reinforcement-learning methods are also increasingly adopted, especially when the objective is to optimize a sequence of decisions, as in algorithmic trading [7].

Standalone ML models can nevertheless have limitations, especially when working with a large number of features [8]. Financial datasets are often characterized by numerous variables, such as historical prices, volume, technical indicators, and sentiment-analysis features. These variables may contain duplicate or irrelevant information that can confuse a model and reduce its precision. A typical ML workflow for stock-price estimation is shown in Figure 1.

In addition, ML algorithms are sensitive to parameter configuration, and finding the best parameter setting can be time-consuming [9]. The unpredictable nature of financial data introduces several challenges that make stock-price prediction an intricate task:

- **Volatility and nonlinearity:** Financial markets are far from stable because of political events, economic policies, and overall investor sentiment.
- **High dimensionality:** Financial datasets may contain stock prices, trading volume, macroeconomic factors, and social sentiment. Such feature spaces often require dimensionality reduction [10].
- **Noisy data:** Financial information is time-varying and may contain noise, outliers, or missing values, all of which negatively affect predictive models.
- **Overfitting:** A large number of attributes can cause a model to learn the training set too closely and fail to generalize to unseen data [11].
- **Real-time prediction:** High-frequency trading requires predictions to be produced accurately and rapidly.

Advanced methodologies address these challenges through data preprocessing, feature selection, and optimization [12]. PCA simplifies a dataset by retaining its most important attributes. Model tuning can employ methods such as Genetic Algorithms (GA) and PSO to improve the accuracy and efficiency of ML models.

Research has also demonstrated that combining multiple ML methods can improve predictive performance. Ensemble methods harness the strengths of their constituent models while minimizing individual weaknesses [13]. For instance, coupling regression models with an optimization method can reduce parameter sensitivity and improve robustness to noisy information.

Information preprocessing occupies an especially important place in contemporary approaches to stock-price pre-

diction [14]. Normalization and scaling prevent features with large ranges from dominating the model, while missing-value imputation addresses absent data.

## 2. RELATED WORK

Forecasting stock prices is complex because of market fluctuation and nonlinearity. Over the years, researchers have applied methodologies ranging from simple statistical models to advanced machine-learning techniques [15]. Related work shows innovation in model development, particularly in integrating heuristic and optimization-based approaches to produce better-performing models.

### 2.1 Conventional Statistical Methods

Basic approaches to stock-price forecasting include ARIMA, ARCH, and GARCH [16]. These techniques decompose time-series data to identify trends. They are useful for portraying linear behavior, but they provide limited solutions for nonlinear and dynamic financial problems in highly volatile markets. This shortcoming motivated researchers to seek methods that respond more effectively to the complexity of financial systems.

### 2.2 Switch to Machine-Learning Models

After the limitations of statistical models became apparent, ML methods were increasingly applied to stock-market forecasting. Artificial Neural Networks (ANNs), SVMs, and Back-Propagation Neural Networks (BPNNs) became effective strategies for capturing nonlinear relationships within financial datasets [17].

### 2.3 Artificial Neural Networks

ANNs mimic aspects of the operation of the brain and are therefore capable of identifying previously unrecognized patterns in financial records. Earlier work showed that they can improve the prediction of stock trends by efficiently managing nonlinear interactions [18]. However, they require large datasets and substantial computation.

### 2.4 Emergence of Hybrid Models

The weaknesses of isolated ML models led researchers to use multi-technique models that combine complementary strengths [19]. Examples include:

- **ANN and Genetic Algorithms:** Hybrid ANNs and GAs can enhance feature selection and stock-index prediction by filtering attributes and removing noise [20].
- **SVM and optimization algorithms:** Integrating SVM with PSO, Ant Colony Optimization, or Teaching-Learning-Based Optimization has proved effective for parameter selection [21].
- **Dimensionality reduction and regression:** PCA, Kernel PCA, and Independent Component Analysis have been incorporated into SVM- and SVR-based regression models to compress the feature space and improve performance [22].
- **Self-Organizing Feature Maps and SVR:** SOFM can cluster and organize input data, after which SVR performs regression. This configuration can improve training speed and prediction rates [23].

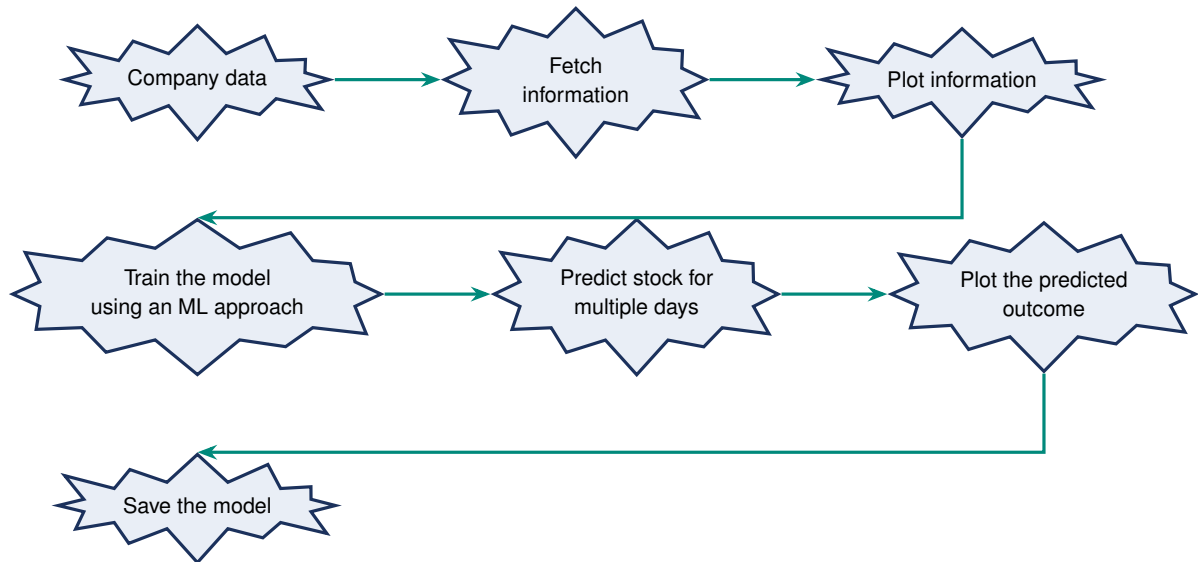


**Figure 1.** Illustrating the dynamics of stock-price estimation using machine learning.

Numerous studies compare the strengths and limitations of different approaches. Accuracy is commonly assessed using MAPE, RMSE, and correlation coefficients. Hybrid models generally provide better accuracy and stability than standalone methods; for example, combining SVM with PSO improves performance through parameter optimization [24]. Models using feature-selection methods such as SOFM and PCA can also achieve higher accuracy by retaining relevant features and discarding irrelevant ones.

Despite significant advances, several research challenges remain [25]:

- **Real-time adaptability:** Many traditional models are not suitable for real-time adaptation in complex financial markets.
- **Cross-market applicability:** More work is needed to determine how well models generalize across different markets.
- **Optimal hybridization:** Selecting the most suitable combination of machine learning, dimensionality reduction, and optimization for a specific financial dataset remains challenging.

The transition from purely statistical techniques to machine learning and hybrid techniques has substantially increased forecast accuracy. Further development should address real-time data processing, interdisciplinary applications, and optimization of hybrid models to provide stronger and more reliable tools for understanding financial markets.

### 3. OBJECTIVE OF THE RESEARCH

The research objectives are:

- **Enhanced predictive accuracy:** Improve stock-price prediction by using powerful ML techniques and optimization algorithms that increase accuracy and dependability in an unpredictable market.
- **Dimensionality reduction:** Apply PCA to make information processing quicker and easier while preserving crucial data.
- **Optimization of model parameters:** Use PSO to tune

SVM parameters and improve performance on complex nonlinear datasets.

### 4. MOTIVATION

Fluctuations and uncertainties in financial markets require sophisticated stock-price prediction methods. Conventional approaches cannot adequately represent nonlinear behavior and high-dimensional spaces. This research seeks to fill these gaps by combining state-of-the-art ML and optimization procedures to produce accurate and relevant information for financial stakeholders.

### 5. PROPOSED WORK

The proposed framework combines PCA, PSO, and SVM to improve stock-price predictability. PCA reduces noise and the number of features relevant to the analysis. SVM is adopted because of its effectiveness when processing non-linearly separable datasets, while PSO further optimizes the SVM hyperparameters. The SVM uses a kernel function to model nonlinear financial data. Together, the three components address high dimensionality, noise, and parameter sensitivity.

#### 5.1 Principal Component Analysis

PCA reduces large sets of observations and variables while preserving the maximum amount of relevant information. It converts a dataset into a new coordinate system in which the first principal component explains the largest variance, the second explains the next largest variance, and so on. PCA forms orthogonal linear combinations of the original features. This facilitates computation, visualization, and analysis of high-dimensional data, and it is commonly used during pre-processing to remove noise.

**Standardize the dataset:** Normalize the values so that the data have zero mean and unit variance:

$$D = \frac{O - \mu}{\sigma}. \quad (1)$$

**Covariance matrix:** Estimate the covariance matrix to un-

derstand correlations between features:

$$\Sigma = \frac{1}{m} D^T D. \quad (2)$$

**Eigen decomposition:** Determine the eigenvalues  $\lambda$  and corresponding eigenvectors  $\mathbf{u}$  of the covariance matrix:

$$\Sigma \mathbf{u} = \lambda \mathbf{u}. \quad (3)$$

**Feature selection:** Select the  $k$  largest eigenvalues and their corresponding eigenvectors as the principal components.

**Data transformation:** Project the standardized data into the lower-dimensional space:

$$B = D U_k, \quad (4)$$

where  $D$  is the standardized dataset,  $O$  is the original information,  $\mu$  is the mean,  $\sigma$  is the standard deviation, and  $U_k$  is the matrix of the top  $k$  eigenvectors.

The suggested framework, illustrated in Figure 2, contains three state-of-the-art algorithms: PCA, SVM, and PSO. PCA extracts important features from the dataset, SVM performs prediction, and PSO tunes the SVM parameters. Model performance is strongly influenced by the kernel type, regularization parameter, and  $\varepsilon$ -insensitive loss. The model uses an RBF kernel because the financial data contain nonlinear relationships.

## 5.2 Particle Swarm Optimization

PSO is a heuristic optimization technique inspired by the behavior of birds or fish moving as a swarm. A population of candidate solutions, called particles, searches for a better solution by updating its position according to its own experience and the knowledge of neighboring particles. Each particle has a velocity calculated using its personal-best and global-best positions. PSO is suitable for nonlinear, multidimensional optimization tasks and does not require gradient evaluation.

**Initialization:** Begin with particles at random positions in the hyperparameter space and with randomly initialized velocities  $a_i$  and  $u_i$ .

**Fitness evaluation:** Evaluate each particle using a fitness function. For SVM regression, the function can be RMSE.

**Personal and global best:** Update each particle's personal best  $q_i$  and the swarm's global best  $g$ .

**Velocity and position:** Update the particle velocity and position according to

$$u_i = w u_i + c_1 s_1 (q_i - a_i) + c_2 s_2 (g - a_i), \quad (5)$$

$$a_i = a_i + u_i. \quad (6)$$

Here,  $w$  is the inertia weight;  $c_1$  and  $c_2$  are cognitive and social acceleration coefficients; and  $s_1, s_2 \in [0, 1]$  are random numbers. Iteration continues until the maximum number of iterations is reached or the global-best fitness is attained.

## 5.3 Support Vector Machine

SVM is a widely used supervised-learning method for classification and regression. Its objective is to determine a hyperplane that maximizes the distance between points from

different classes in a high-dimensional feature space. For regression, Support Vector Regression (SVR) introduces an error tolerance within a specified range. SVM performs well on nonlinear problems by using kernel functions such as the RBF kernel to map data into a higher-dimensional space.

The RBF kernel used for nonlinear relationships is

$$I(a, a') = \exp(-\gamma \|a - a'\|^2). \quad (7)$$

To balance model complexity and prediction error, SVM minimizes

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{k=1}^m (\xi_k + \xi_k^*), \quad (8)$$

subject to

$$b_i - (\mathbf{w} \cdot \phi(a_i) + y) \leq \varepsilon + \xi_k, \quad (9)$$

$$(\mathbf{w} \cdot \phi(a_i) + y) - b_i \leq \varepsilon + \xi_k^*, \quad (10)$$

$$\xi_k, \xi_k^* \geq 0. \quad (11)$$

After optimization, the prediction for an independent point  $a$  is

$$\hat{b} = \mathbf{w} \cdot \phi(a) + y. \quad (12)$$

Here,  $\gamma$  adjusts the influence of each training point;  $C$  is the penalty parameter;  $y$  is the bias;  $\mathbf{w}$  is the weight vector;  $\phi(a)$  is the kernel-transformed input;  $\varepsilon$  is the loss parameter;  $\xi_k$  and  $\xi_k^*$  are slack variables; and  $\hat{b}$  is the predicted stock price.

## 5.4 Algorithm for PCA-PSO-SVM

**Input:** Dataset  $E$  with features  $a$  and labels  $b$ . Parameters for PCA ( $k$ ), PSO ( $w, c_1, c_2$ ), and SVM.

1. **Data preprocessing:** Normalize  $a$ ; handle missing values and outliers.
2. **Dimensionality reduction:** Compute  $U_k$ , the top  $k$  eigenvectors, and transform  $a$  using  $U_k$ .
3. **Initialize PSO:** Select random particle positions  $a_i$  and velocities  $u_i$  in the hyperparameter space.
4. **Optimize SVM parameters:** At every iteration, evaluate particle fitness using RMSE, update  $q_i$  and  $g$ , and update  $u_i$  and  $a_i$ .
5. **Train SVM:** Apply the optimized  $C$ ,  $\gamma$ , and  $\varepsilon$  to the reduced dataset.
6. **Test the model:** Evaluate the final model built with the optimized SVM hyperparameters.

The proposed PCA-PSO-SVM approach improves stock-price prediction by eliminating multicollinearity, optimizing hyperparameters, and exploiting SVM's ability to represent nonlinear relationships. The integrated framework improves forecast breadth, depth, and adaptability, making it suitable for modeling price movements in intricate stock markets.

## 6. RESULTS

The results demonstrate the high accuracy of the proposed PCA-PSO-SVM framework for forecasting stock prices. PCA is used for dimensionality reduction, PSO for hyperparameter optimization, and SVM regression for modeling

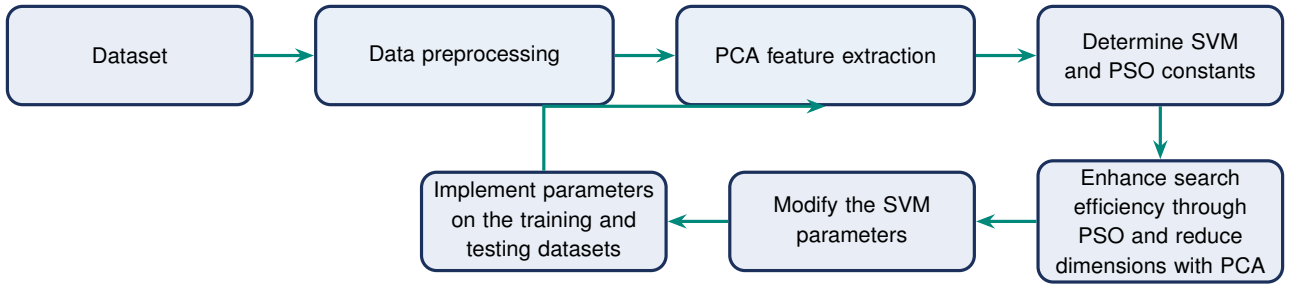


**Figure 2.** An integrated method utilizing PCA, PSO, and SVM.

high-dimensional, noisy financial information. RMSE and MAPE are used to assess prediction capability. The findings show that the developed model provides better prediction efficiency than simple statistical models and standalone ML algorithms.

**Accuracy (ACC):** The ratio of correct forecasts to all forecast stock-price movements:

$$\text{Accuracy} = \frac{CP + CN}{CP + CN + IP + IN}, \quad (13)$$

where  $CP$ ,  $CN$ ,  $IP$ , and  $IN$  denote correct positive, correct negative, incorrect positive, and incorrect negative predictions, respectively.

**Precision (Pre):** The ratio of correct positive predictions to all positive predictions:

$$\text{Precision} = \frac{CP}{CP + IP}. \quad (14)$$

**Recall (Rec):** The proportion of actual positives identified correctly:

$$\text{Recall} = \frac{CP}{CP + IN}. \quad (15)$$

**Sensitivity (Sen):** The ability to detect a positive stock-price movement:

$$\text{Sensitivity} = \frac{CP}{CP + IN}. \quad (16)$$

**Specificity (Spe):** The ability to forecast a decline in stock price correctly:

$$\text{Specificity} = \frac{CN}{CN + IP}. \quad (17)$$

A correct positive is a day on which the stock price rose and the model anticipated the rise. A correct negative is a day on which the price declined and the model anticipated the decline. An incorrect positive is a predicted increase that did not occur, while an incorrect negative is a missed increase.

**Mean Absolute Error (MAE):** The average magnitude of the errors, irrespective of direction:

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^m |b_i - \hat{b}_i|. \quad (18)$$

**Root Mean Square Error (RMSE):** The square root of the

mean squared difference between forecast and actual values:

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^m (b_i - \hat{b}_i)^2}. \quad (19)$$

**Mean Absolute Percentage Error (MAPE):** The mean absolute error relative to the actual values:

$$\text{MAPE} = \frac{100}{m} \sum_{i=1}^m \left| \frac{b_i - \hat{b}_i}{b_i} \right|. \quad (20)$$

Here,  $m$  is the number of observations,  $b_i$  is the actual stock price at observation  $i$ , and  $\hat{b}_i$  is the predicted stock price.

**Table 1.** Performance comparison of existing approaches with the suggested approach.

| Approach    | Accuracy (%) | Precision (%) | Recall (%) | Sensitivity (%) | Specificity (%) |
|-------------|--------------|---------------|------------|-----------------|-----------------|
| SVM         | 84.5         | 83.2          | 84.1       | 84.5            | 84.3            |
| PSO-SVM     | 86.5         | 85.8          | 86.11      | 86.2            | 85.9            |
| PCA         | 83.2         | 82.1          | 82.3       | 82.1            | 83.1            |
| PCA-SVM-PSO | 95.5         | 95.1          | 95.3       | 95.2            | 95.1            |

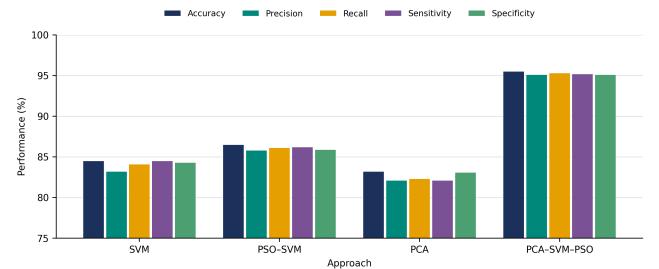
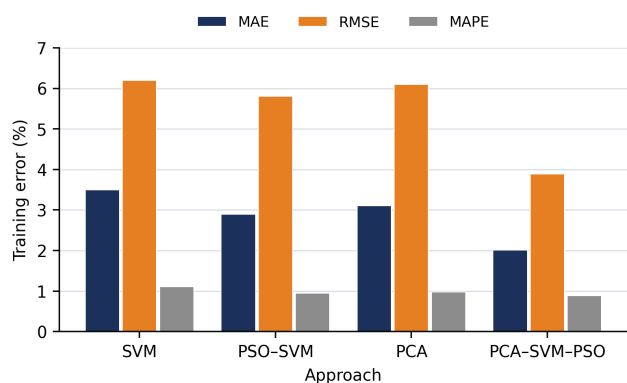


**Figure 3.** Graphical representation of the performance comparison of existing approaches with the suggested approach.

Table 1 and Figure 3 compare the proposed PCA-SVM-PSO approach with related methods. The proposed approach achieves an accuracy of 95.5%, compared with 84.5% for standalone SVM, 86.5% for PSO-SVM, and 83.2% for PCA. Its precision, recall, sensitivity, and specificity are all above 95%, indicating that the model minimizes both false-positive and false-negative predictions. By combining PCA for dimensionality reduction, PSO for hyperparameter optimization, and SVM for regression, the study produces a stable framework that handles data complexity and parameter tuning more effectively.

**Table 2.** Training-phase error rates for existing approaches and the suggested approach.

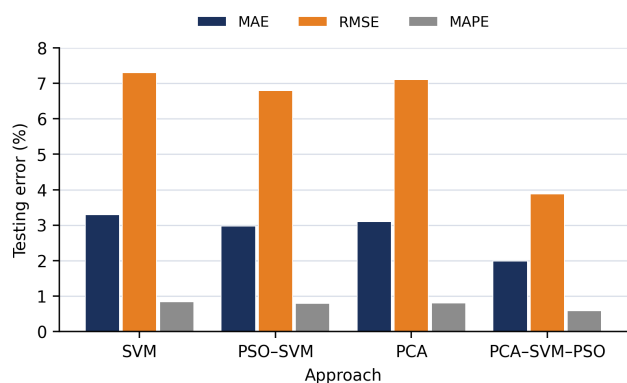
| Approach    | MAE (%) | RMSE (%) | MAPE (%) |
|-------------|---------|----------|----------|
| SVM         | 3.50    | 6.20     | 1.11     |
| PSO-SVM     | 2.89    | 5.80     | 0.95     |
| PCA         | 3.10    | 6.10     | 0.98     |
| PCA-SVM-PSO | 2.01    | 3.89     | 0.88     |

**Figure 4.** Graphical comparison of training-phase error rates for existing approaches and the suggested approach.

The training errors in Table 2 and Figure 4 show that PCA-SVM-PSO produces the lowest values: MAE of 2.01%, RMSE of 3.89%, and MAPE of 0.88%. PSO-SVM, with MAE of 2.89% and RMSE of 5.8%, performs better than standalone SVM. The combined method benefits from PCA's dimensionality reduction, SVM's robust regression capability, and PSO's optimization of SVM hyperparameters.

**Table 3.** Testing-phase error rates for existing approaches and the suggested approach.

| Approach    | MAE (%) | RMSE (%) | MAPE (%) |
|-------------|---------|----------|----------|
| SVM         | 3.30    | 7.30     | 0.84     |
| PSO-SVM     | 2.97    | 6.80     | 0.79     |
| PCA         | 3.10    | 7.10     | 0.81     |
| PCA-SVM-PSO | 1.99    | 3.88     | 0.59     |

**Figure 5.** Graphical comparison of testing-phase error rates for existing approaches and the suggested approach.

The testing-error comparison in Table 3 and Figure 5 highlights the superior performance of PCA-SVM-PSO, which

achieves the lowest errors across all metrics: MAE of 1.99%, RMSE of 3.88%, and MAPE of 0.59%. This indicates strong out-of-sample applicability. The standalone SVM and PSO-SVM methods exhibit higher errors, while PCA provides moderate improvement. Integrating PCA for dimensionality reduction, SVM for regression, and PSO for hyperparameter selection prevents significant errors during testing.

## 7. CONCLUSION AND FUTURE ENHANCEMENT

Fluctuations in stock prices are difficult to explain because of the variety of global, national, and market factors involved. Forecasting is challenging in a dynamic field where variables change unexpectedly. Conventional prediction techniques have limitations, especially when hidden nonlinear associations and increased data dimensionality are present. Machine learning supported by optimization techniques provides a promising solution.

This study presents a comprehensive investigation of stock-price fluctuation by establishing an ensemble model consisting of PCA, PSO, and SVM. PCA makes it possible to work efficiently with many features and high-dimensional datasets. PSO tunes the SVM hyperparameters to improve predictive efficiency, while SVM captures nonlinear relationships and makes the framework more resilient to noise and volatility. By reducing data complexity, filtering noise, and handling parameter sensitivity, the combined forecasting approach improves prediction accuracy by a wide margin. The work contributes a stable, high-performing approach to stock-price prediction and financial analysis.

Potential enhancements include adding more powerful optimization algorithms, such as genetic algorithms; using real-time data and sentiment analysis; and employing deeper architectures, such as LSTMs, to model temporal dependencies. Applying the framework to other financial assets, including cryptocurrencies and commodities, may further improve its flexibility.

## REFERENCES

- [1] L. Nemes and A. Kiss, "Prediction of stock values changes using sentiment analysis of stock news headlines," *Journal of Information and Telecommunication*, vol. 5, no. 3, pp. 375–394, 2021.
- [2] H. S. Salih and F. A. Mohamed, "Fusion-based diversified model for internet of vehicles: Leveraging artificial intelligence in cloud computing," *Fusion: Practice and Applications*, vol. 12, no. 2, pp. 54–69, 2023.
- [3] Z. N. Al-kateeb and D. B. Abdullah, "A smart architecture leveraging fog computing fusion and ensemble learning for prediction of gestational diabetes," *Fusion: Practice and Applications*, vol. 12, no. 2, pp. 70–87, 2023.
- [4] V. Gupta and M. Ahmad, "Stock price trend prediction with long short-term memory neural networks," *International Journal of Computational Intelligence Studies*, vol. 8, no. 4, pp. 289–305, 2019.
- [5] X. Zhang, S. Qu, J. Huang, B. Fang, and P. Yu, "Stock market prediction via multi-source multiple instance learning," *IEEE Access*, vol. 6, pp. 50 720–50 728, 2018.
- [6] M. Elsharkawy and I. S. Farahat, "A proposed predictive model for business telemarketing information management," *Journal*

- of *Cybersecurity and Information Management*, vol. 9, no. 1, pp. 27–39, 2021.
- [7] L. Wang, F. Ma, J. Liu, and L. Yang, “Forecasting stock price volatility: New evidence from the GARCH-MIDAS model,” *International Journal of Forecasting*, vol. 36, no. 2, pp. 684–694, 2020.
- [8] L. Yang and T. Zhai, “Research on sentiment tendency analysis of video reviews based on sentiment dictionary,” *Network Security Technology and Application*, vol. 255, no. 3, pp. 53–56, 2022.
- [9] A. E. Khedr, S. Salama, and N. Yaseen, “Predicting stock market behavior using data mining technique and news sentiment analysis,” *International Journal of Intelligent Systems and Applications*, vol. 9, no. 7, pp. 22–30, 2017.
- [10] M. N. Dawood, M. A. Alkhafaji, A. Hussian, H. A. Diame, N. A. Hussien, S. Yassine, and V. Rajinikanth, “Developing a risk management system with an optimistic predictive approach and business decision-making,” *Journal of Intelligent Systems and Internet of Things*, vol. 9, no. 2, pp. 51–64, 2023.
- [11] A. Sariga and J. Uthayakumar, “Type 2 fuzzy logic based unequal clustering algorithm for multi-hop wireless sensor networks,” *International Journal of Wireless and Ad Hoc Communication*, vol. 1, no. 1, pp. 33–46, 2020.
- [12] I. V. Pustokhina, “Blockchain technology in the international supply chains,” *International Journal of Wireless and Ad Hoc Communication*, vol. 1, no. 1, pp. 16–25, 2020.
- [13] K. Parvesh, C. Tharun, and M. Prakash, “Apparel recommendation engine using inverse document frequency and weighted average Word2vec,” *Journal of Cognitive Human-Computer Interaction*, vol. 1, no. 2, pp. 46–56, 2021.
- [14] K. Vijay, “Collaborating the textual reviews of the merchandise and foretelling the rating supported social sentiment,” *Journal of Cognitive Human-Computer Interaction*, vol. 1, no. 2, pp. 63–72, 2021.
- [15] Y. Trichilli, M. B. Abbes, and A. Masmoudi, “Predicting the effect of googling investor sentiment on islamic stock market returns: A five-state hidden Markov model,” *International Journal of Islamic and Middle Eastern Finance and Management*, vol. 13, no. 2, pp. 165–193, 2020.
- [16] V. Roy *et al.*, “Network physical address based encryption technique using digital logic,” *International Journal of Scientific & Technology Research*, vol. 9, no. 4, pp. 3119–3122, 2020.
- [17] S. V. Kolasani and R. Assaf, “Predicting stock movement using sentiment analysis of Twitter feed with neural networks,” *Journal of Data Analysis and Information Processing*, vol. 8, no. 4, pp. 309–319, 2020.
- [18] Y. Han, I. M. El-Hasnony, and W. Cai, “Dragonfly algorithm with gated recurrent unit for cybersecurity in social networking,” *Journal of Cybersecurity and Information Management*, vol. 0, no. 2, pp. 75–88, 2019.
- [19] E. Mohamed, “The relationship between artificial intelligence and internet of things: A quick review,” *Journal of Cybersecurity and Information Management*, vol. 1, no. 1, pp. 30–34, 2020.
- [20] M. M. Rounaghi and F. N. Zadeh, “Investigation of market efficiency and financial stability between S&P 500 and london stock exchange: Monthly and yearly forecasting of time-series stock returns using ARMA model,” *Physica A: Statistical Mechanics and its Applications*, vol. 456, pp. 10–21, 2016.
- [21] M. Vijn, D. Chandola, V. A. Tikkiwal, and A. Kumar, “Stock closing price prediction using machine learning techniques,” *Procedia Computer Science*, vol. 167, pp. 599–606, 2020.
- [22] Y. Ji, A. W. Liew, and L. Yang, “A novel improved particle swarm optimization with long short-term memory hybrid model for stock indices forecast,” *IEEE Access*, vol. 9, pp. 23 660–23 671, 2021.
- [23] G. Ataman and S. Kahraman, “Stock market prediction in BRICS countries using linear regression and artificial neural network hybrid models,” *Singapore Economic Review*, vol. 67, no. 2, pp. 635–653, 2022.
- [24] V. Roy, “An improved image encryption consuming fusion transmutation and edge operator,” *Journal of Cybersecurity and Information Management*, vol. 8, no. 1, pp. 42–52, 2021.
- [25] H. Liu and Y. Hou, “Application of bayesian neural network in prediction of stock time series,” *Computer Engineering and Applications*, vol. 55, no. 12, pp. 225–229, 2019.