International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 3, June 2023, pp. 3549~3560
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i3.pp3549-3560

Analysis of Nifty 50 index stock market trends using hybrid machine learning model in quantum finance

Chinthakunta Manjunath[1], Balamurugan Marimuthu[1], Bikramaditya Ghosh[2]
[1]Department of Computer Science and Engineering, School of Engineering and Technology, CHRIST (Deemed to be University), Bengaluru, India
[2]Symbiosis Institute of Business Management, Symbiosis International (Deemed University), Bengaluru, India

Article Info

Article history:

Received Jul 14, 2022
Revised Sep 20, 2022
Accepted Oct 1, 2022

Keywords:

National stock exchange fifty
Principle component analysis
Stock market
Technical indicators
Time series forecast

ABSTRACT

Predicting equities market trends is one of the most challenging tasks for market participants. This study aims to apply machine learning algorithms to aid in accurate Nifty 50 index trend predictions. The paper compares and contrasts four forecasting methods: artificial neural networks (ANN), support vector machines (SVM), naive bayes (NB), and random forest (RF). In this study, the eight technical indicators are used, and then the deterministic trend layer is used to translate the indications into trend signals. The principal component analysis (PCA) method is then applied to this deterministic trend signal. This study's main influence is using the PCA technique to find the essential components from multiple technical indicators affecting stock prices to reduce data dimensionality and improve model performance. As a result, a PCA-machine learning (ML) hybrid forecasting model was proposed. The experimental findings suggest that the technical factors are signified as trend signals and that the PCA approach combined with ML models outperforms the comparative models in prediction performance. Utilizing the first three principal components (percentage of explained variance=80%), experiments on the Nifty 50 index show that support vector classifier (SVC) with radial basis function (RBF) kernel achieves good accuracy of (0.9968) and F1-score (0.9969), and the RF model achieves an accuracy of (0.9969) and F1-Score (0.9968). In area under the curve (AUC) performance, SVC (RBF and Linear kernels) and RF have AUC scores of 1.

Corresponding Author:

Chinthakunta Manjunath
Department of Computer Science and Engineering, School of Engineering, CHRIST (Deemed to be University)
Mysore Road, Kengeri Campus, Kumbalgodu, Bengaluru
Email: manju.chintell@gmail.com

1. INTRODUCTION

Predicting financial market trends has attracted a lot of researchers for several decades. It can be divided into the fundamental analysis method, which uses the company's critical information like dividend value, P/E, P/B, market position, expenditures, yearly growth rates, and the technical analysis method, which essences on prior equity prices [1]. The traditional statistical approaches, including logistic regression, exponential average, autoregressive integrated moving average (ARIMA), and generalized autoregressive conditionally heteroscedastic (GARCH), were employed to forecast equity market price [2], [3]. Traditional time series models, for example, generally handle linear forecasting models, and variables must follow a statistically normal distribution. On the other hand, statistical approaches assume that a linear process forms

the sequence data and performs poorly in non-linear stock price movement predictions. Hence, machine learning (ML) and deep learning (DL) approaches are gradually being explored in price changes in stock market predictions due to their success in non-linear financial data [4]. Technical analysis is one of the most extensively utilized feature extraction analyses for predicting the financial market, leading to improved projections [5].

In financial applications, ML techniques have been effectively used on financial data due to their capacity to fit and forecast performance for complicated data sets [6]–[9]. The hybrid financial time-series model is proposed for forecasting stock prices based on a feature selection approach using a genetic algorithm by maximizing the support vector regression parameters [10]. The effect of the principal component analysis (PCA) technique for decreasing data dimensionality was discovered and it is used to investigate financial time series, develop dynamic trading techniques, and calculate financial risk [11]. A stock-selection model is built using a machine-learning algorithm that can conduct non-linear stock categorization [12]. To forecast financial time series, they utilize PCA to derive low-dimensional efficient data and demonstrated that PCA was created to estimate daily stock market returns [13]–[15].

The Istanbul stock exchange (ISE) 100 Index prices are forecasted using artificial neural networks (ANN) and support vector machines support vector machines (SVM) models. Using ten technical indicators as inputs, the extreme 75.74% and 71.52% accuracy in ANN and polynomial kernel SVM were obtained. The authors did not explain why their model performed better than other models in earlier publications, which is one of the work's flaws [16]. Extracting useable features from various monetary parameters is one of the most significant and challenging aspects of equity price movement prediction. technical analysis (TA) is employed to obtain market characteristics from the financial data, and equity market prediction frequently utilizes this analysis to provide significant features for machine learning models [17]. The six worldwide equity markets, including established and emerging economies, were investigated to anticipate stock returns. The hit ratio of the Nifty index by prior closing earnings achieved 51.0% using the SVM machine learning model [18]. Nonparametric machine learning methods such as ANN and SVM with polynomial and radial basis function kernels are used to anticipate the trends of the Korea composite stock price index 200 (KOSPI 200) prices using technical analysis. These three models had 50.23%, 49.43%, and 52.90% accuracy, respectively [1].

The researchers discovered that adapting technical indications into deterministic drift signals enhances prediction accuracy and made forecasts for Nifty and Sensex indexes using ANN, SVM, RF, and NB machine learning models. However, the authors did not perform dimensionality reduction [19]. Harmony search and GA were employed to improve a standard ANN model before using improved ANN to make a prediction. The prediction performance is enhanced by combining the model with different strategies. And the findings revealed that, compared to the other models, the suggested ANN model is the most dominating [20]. A flexible SVR was created for equity data over a range of time durations. According to the results, the modified two-stage fusion SVR outperformed the traditional SVR when learning parameters were dynamically optimized using PSO [21]. Integration of SVMs and an ARIMA model in statistical modeling allows the *r* replication of both linear and non-linear characteristics. In the suggested hybrid model, the ARIMA residuals were modeled SVMs [22]. Trend forecasting in financial markets was performed using four hybrid classifiers and the performance of these classifiers outperformed that of separate proximal SVMs, and the SVM with random forests outperformed all other prediction approaches [23].

It was evident from the preceding study background that each algorithm can successfully tackle equity market prediction issues. However, it is essential to note that each has its restrictions in terms of its appropriate feature selection, model building with hyperparameter optimization, and acceptable results in previous work. The depiction of the input data impacts the forecast outcomes, as does the prediction technique. Furthermore, instead of utilizing all characteristics as input data, employing just prominent features and recognizing them as input data will increase the forecasting model’s correctness. Therefore, this research presents a realistic and workable hybrid framework for investors. This study uses a hybrid machine learning model (PCA-ML) that uses deterministic trend data to estimate the Nifty 50 index price trend more accurately than previous techniques. In a nutshell, the primary offerings of this study are as follows: i) this paper suggests comprehensive feature engineering and a customized PCA-ML model were developed to predict the trends of the Nifty 50 index, ii) the proposed model employs PCA to identify the critical components from several technical indicators that impact stock prices to minimize data dimensionality and enhance model performance, and iii) the experimental findings suggest that the proposed hybrid model (PCA approach combined with ML models) outperforms the comparative models in prediction performance.

The remaining sections of the article are as follows. The following section 2 covers data descriptions, methodology and PCA, and four classifier techniques. Extensive experimental and analytical results supporting the proposed model are presented in section 3. Section 4 presents the conclusion of the research.

2. METHOD AND PREDICTION MODELS

In the proposed work, technical analysis and machine learning methods are combined to evaluate the Indian National Stock Exchange. This section presents the datasets and their mathematical equations, purpose, data labeling, the comprehensive architecture of the suggested model. PCA and ML techniques are also presented.

2.1. Research data and data labeling

This analysis uses ten years of daily Nifty 50 index data from January 1, 2011, through December 31, 2019, for nine years provided by [24]. it consists of 2,226 total samples with open (*o*), close (*c*), high (*h*), low (*l*), and traded shares (*V*) values. After pre-processing and technical analysis feature engineering, the data consists of 2.140 samples. The equity market trend is classified into two classes in finical markets up and down, it indicates the equity price movement. In this study, the labels are derived using the daily close price of a stock market index using (1). Let C_t be the stock index's close price on day *t*. The t^{th} day's class label is specified as using (1).

$$target_t = \begin{cases} 1, & C_{t+1} > C_t, \\ 0, & Otherwise. \end{cases} \quad (1)$$

Technical indicators (TI) are well-known for predicting the equity market and these indicators are simple mathematical models based on open and closing prices. This article applied eight technical indicators based on the previous studies [25] and informed them about deterministic drift values [19] before inputting them into the proposed hybrid prediction model. These TI are illustrated in Table 1.

Table 1. The formulas for technical indicators

Feature Name	Formula	Purpose of each of the technical data and its trend deterministic signal
Simple moving average (SMA)	$SMA_n = \frac{1}{n}\sum_{i=0}^{n-1} C_{t-i}$	SMA: The SMA is a calculation of a security's closing price averaged over time, and it uses ten days SMA values are used. If $C_t > SMA_t$, then set "1"; else set "-1."
Exponential moving average (EMA)	$EMA = (P \times \alpha) + ((Previous\ EMA \times (1-\alpha))$ $\alpha = Smoothing\ Factor = \frac{2}{1+N}$	EMA: The EMA represents the average price and gives recent prices greater weight. It uses ten days SMA values are used. If $C_t > EMA_t$, then set "1"; else set "-1."
Moving average convergence divergence (MACD)	$MACD = 12Peroid\ EMA - 26Period\ EMA$	MACD: The gap between two variable exponential moving averages is (MACD). They are used for the buying and selling of signals. If $MACD_t > MACD_{t-1}$, then set "1"; else set "-1".
Relative strength index (RSI)	$100 - \frac{100}{1 + (\sum_{i=0}^{n-1} UP_{t-i}/n)/(\sum_{i=0}^{n-1} DW_{t-i}/n)}$	RSI: If the RSI falls below 30 or over 70, it indicates that the market is oversold or overbought, respectively. It might be anywhere from 0 to 100. If $RSI_t <= 30$ or $RSIt > RSI_{t-1}$ then set "1"; and if $RSI_t >= 70$ or $RSIt < RSI_{t-1}$ then set "-1"; if RSI is between 30 and 70 if RSI at a time 't' is > RSI at a time 't-1', then label '1'; else label '-1'.
Stochastic momentum index (SMI)	$SMI = \frac{C_t - LL_{t-(n-1)}}{HH_{t-(n-1)} - LL_{t-(n-1)}} \times 100$	Stochastic Momentum Index (SMI): The SMI will determine the final price using the high-low range average. If SMI >-50, then label "+1". If SMI <50, then label "-1".
Relative volatility index (RVI)	$ŕ = 100\ ŭ / (ŭ + \delta)$	Relative volatility index (RVI): The RVI has been used to determine the direction of equity value uncertainty. The RVI value of '50' has been set aside as a significant point, and if the RVI is greater than the important point, the label "1"; otherwise, label "-1."
Ease of movement (EMV)	$EMV = \frac{Distance}{Box\ ratio}$	Ease of movement (EMV): It shows how quickly the value of an asset can increase or decrease in response to its trading volume. If EMV > 0 then set "1"; else set "-1".
Reynolds number (RN)	$RN = RVI/EMV$ Or $R_e = (100\ ŭ / (ŭ + \delta))\ / (ϱ / Ч)$	Reynolds number (RN): The Reynolds number, a theory from fluid mechanics that is centuries old, has been adapted in the field of investment finance to describe potentially explosive market situations. If RN is less than 3.142, set "1"; if RN is more than 3.142, set "-1"; [26]–[28].

While, C_t is the closing price at time t, and n is an input window length. P=Current Price and N = Number of Time Periods. UP_t means upward price change while DW_t is the downward price change at time t respectively;
LL = Lowest Low and HH= Highest High. RVI = ŕ. ŭ – Wilder's smoothing of USD, and δ – Wilder's smoothing of DSD.

$Distance = ϱ = \frac{High+Low}{2} - \frac{Prior\ High+Prior\ Low}{2}$, $BoxRatio = Ч = \frac{\frac{Volume}{100,000,000}}{High-Low}$

2.2. Architecture

The architecture of our suggested model is depicted in Figure 1, which is divided into three phases: financial input data representation, PCA for dimensional reduction, and classification models for trend prediction. The proposed hybrid model fulfills these goals. Time series data for financial technical indicators are constructed via feature engineering, and then the trend deterministic data preparation layer translates this data into an upward (+1) or downward (-1) trend. The trend deterministic data preparation layer output is then sent into the PCA method. To minimize data dimensionality and boost model performance, this PCA method isolates the most important factors from a wide range of technical indicators that impact stock prices. Finally, four ML models are employed to assess the effectiveness of the forecasting model by utilizing the main components of high-weighted variables.

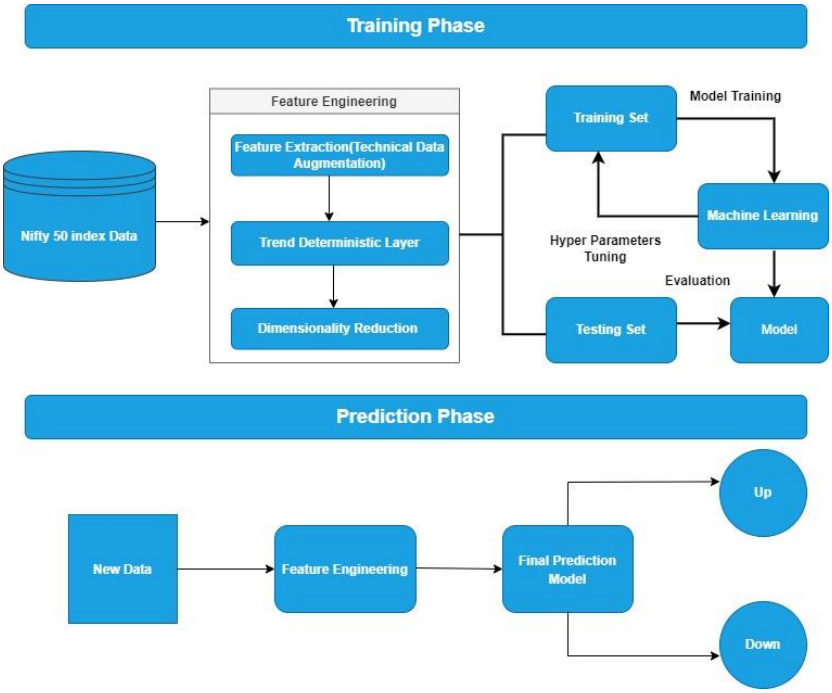


Figure 1. The architecture of our suggested hybrid model

2.3. Prediction models

2.3.1. Principal component analysis

principal component analysis (PCA) is a multivariate statistical technique and unsupervised machine learning algorithm that reduces variables by eliciting important evidence from data, affects the data matrix to conserve as much evidence as possible, and generates the primary pattern from data. PCA tries to reduce the dimensionality of a data set with many related variables while maintaining the data set's maximum variability. As a result, the first principal component retains the greatest variation from the original characteristics [29], [30]. The effect of the PCA technique for decreasing data dimensionality was discovered and it is used to investigate financial time series, develop dynamic trading techniques, and calculate financial risk [11].

2.3.2. Artificial neural networks model

Artificial neural network (ANN) is a dense network of interconnected neurons that is triggered by inputs. The transfer function of a single neuron in the output layer is log sigmoid. Figure 2 illustrates ANN's design. In this experiment, the neural network's inputs are the technical indicators from Table 1. A threshold of 0.5 determines whether the movement will be up or down. If the output value is greater than or equal to 0.5, the prediction is an uptrend. Otherwise, it is a downward trend [31]. The link between weights, biases, and nodes is shown in (2). The weighted total of inputs is transferred from one layer to another by a non-linear activation function. It may be thought of as a vector, with n representing the amount of data for the final node, f representing the activation function, x_1, x_2... and x_n representing the inputs, w_1, w_2 and w_n representing the weights, and z representing the final output.

$$z = f(x \cdot w + b) = f(\sum_{i=1}^{n} x_i w_i + b) \quad (2)$$

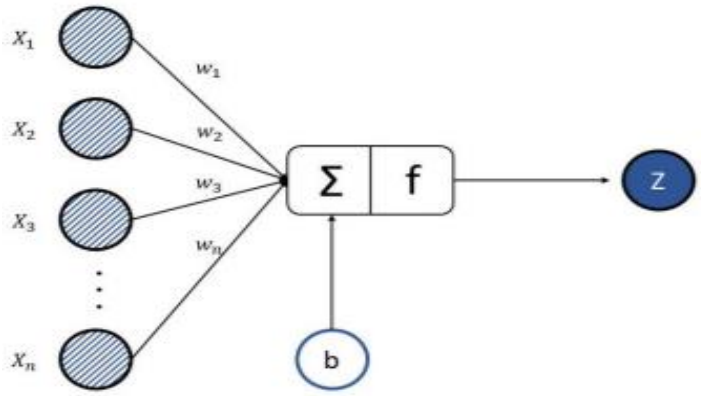


Figure 2. The link between ANN inputs and outputs is depicted in this diagram

2.3.3. Support vector machine

Support vector machine (SVM) is a non-linear method used for classification and regression modeling. The core idea is to map characteristics in a larger-dimensional space nonlinearly. In this area, a hyperplane is created to establish class borders. A kernel function such as a radial basis, a sigmoidal, or a polynomial function should be used for the best hyperplane separation [23]. The kernel function is a significant asset of the algorithm. SVM can be used to solve binary as well as multiclass issues. The primary aim of an SVM is to find the hyperplane with the most significant margin [29], [32]. The concept is to increase the difference between positive and negative instances. The ultimate decision boundary determines the most critical margin hyperplane. SVMs may map input vectors $x_i \in R^d$ into a high-dimensional feature space$\Phi(x_i) \in H$, which is mapped by a kernel function $K(x_i x_j)$. The SVM defines the decision boundary in (3), and the SVM method is demonstrated in Figure 3 [32].

$$f(x) = sgn(\sum_{i=1}^{n} \alpha_i y_i \cdot K(x, x_i) + b) \quad (3)$$

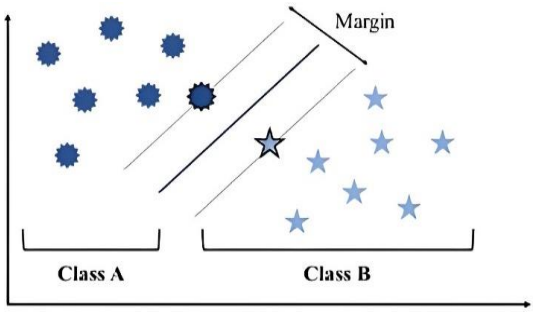


Figure 3. Illustration of the SVM method

The kernel function formula is shown in (4) to (6), where *d* is the degree of a polynomial function and is the radial basis function constant. The slope and the intercepted constant, abbreviated *c*, are two variables that can be changed for the sigmoid function.

$$RBF: K(x_i, x_j) = exp\ (-y\ ||x_i - x_j||^2) \quad (4)$$

$$Polynomial: K(x_i, x_j) = (x_i \cdot x_j + 1)^d \quad (5)$$

$$Sigmoid: K(x_i, x_j) = \tanh\ (\alpha x_i^T y + c) \quad (6)$$

2.3.4. Naïve bayes algorithms

The supervised naive Bayes method is based on the Bayesian theorem and the idea that the features of the dataset are uncorrelated (independent). Bayes' theorem (7), where y is the class variable and x_1 through x_n are dependent feature vectors, expresses this relationship. Additionally, it assumes that hidden or latent factors have no impact on the predictions (thus the name "nave") [31].

$$P(y|x_1x_2 \dots x_n) = P(y)\frac{\prod_{i=1}^{n} P(x_i|y)}{P(x_1x_2 \dots x_n)} \quad (7)$$

2.3.5. Random forests

Another technique to avoid overfitting is to integrate the predictions of several different models into a single forecast, which is commonly done via a plurality vote in classification and averaging in regression [31]. Ensemble learning is a broad term for this method. Consider a collection of uncorrelated random variables $\{y_i\}_{i=1}^{n}$ with an expected mean $\mathrm{E}[Y_i] = \mu$ and $r(Y_i) = \sigma^2$ to illustrate why averaging is used. The expectation for the average of these is the same it is shown in (8).

$$\mathrm{E}\left[\frac{1}{n}\sum_{i=1}^{n} Y_i\right] = \frac{1}{n}\sum_{i=1}^{n} E[Y_i] = \frac{1}{n} \cdot n\mu = \mu \quad (8)$$

but reduced variance compared to each of the individuals Y_i s: depicted in (9).

$$Var\left(\frac{1}{n}\sum_{i=1}^{n} Y_i\right) = \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} Var(Y_i) = \frac{1}{n^2} \cdot n\,\sigma^2 = \frac{\sigma^2}{n} \quad (9)$$

In the framework of ensemble methods, these Y_i are equivalent to the forecast made by classifier i. The projected value of the collective forecast is the same as any individual forecast, with a tiny difference.

3. EXPERIMENTAL RESULTS AND DISCUSSION

This section discusses the experimental findings as well as the assessment metrics used in the suggested prediction framework. All experiments in this study were carried out using Python 3 along with Scikit Learn and the Keras package. To train the model and assess performance, the primary dataset is split into a training set (70% of samples) and a testing set (30% of samples). Summary statistics of the Nifty 50 index selected indicators are displayed in Table 2. A common statistic used to assess the severity of a problem with binary classification is the ROC curve. Classification evaluation metrics such as F1-score, accuracy, and receiver operating characteristics area under the curve (ROC-AUC) are used to assess the quality of ML models and their predictions. To calculate the F1 score and the Accuracy, we must first analyze the precision and recall in terms of the true positive (TP), true negative (TN), false positive (FP), and false negative (FN). In Table 3, we have a compilation of all of these measurements.

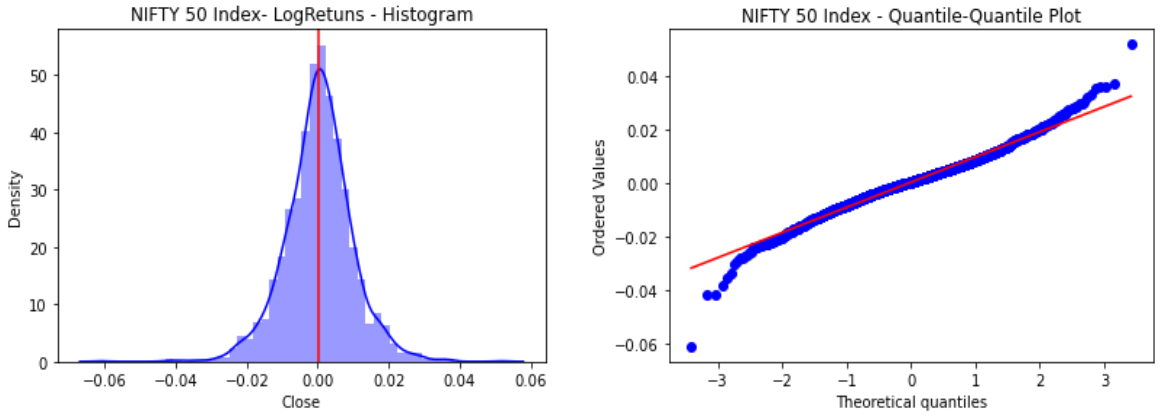
Table 2. Summary statistics of the Nifty 50 index selected indicators

Technical Indicator	Mean	Standard deviation
SMA	8070.973292	2158.820072
EMA	8078.466271	2162.284020
MACD	20.440753	78.590860
SMI	80.817425	198.108590
RSI	53.768689	12.351120
EMV	838.557251	1034.583458
RVI	53.768689	12.351120
RN	0.536157	4.780517

Table 3. Metrics for evaluation of the model

Metric Name	Formula
Accuracy	$Accuarcy = \frac{(TN + TP)}{(TN + TP + FN + FP)}$
F1-Score (F1 Measure)	$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall}$
Precision	$Precision = \frac{TP}{TP + FP}$
Recall	$Recall = \frac{TP}{TP + FN}$

Figure 4 shows the outcome of the Nifty 50 index log returns in Figure 4(a), and its Q-Q plot in Figure 4(b). Table 4 shows the selection of the best hyperparameters using GridSearchCV. These hyperparameters regulate the training process and greatly affect the effectiveness of machine learning models. In our study, we provided four ML models for experimentation, and we used GridsearchCV to pick the best hyperparameters for each. The experimental results demonstrate that improved performance was attained by employing these optimal hyperparameters while employing the GridearchCV method.



(a) (b)

Figure 4. Illustration of (a) Nifty 50 index log returns and (b) Nifty 50 index Q-Q plot

Table 4. Best hyperparameters selection using GridSearchCV

ANN Model		RF Model	
Parameter Name	Value	Parameter Name	Value
epochs	10	bootstrap	True
batch_size	32	max_depth	10
learning_rate	0.1	max_features	auto
Number of hidden layers	25	min_samples_leaf	1
activation	ReLU	min_samples_split	6
optimizer	RMSprop	n_estimators	50
GNB Model		SVC Model	
Parameter Name	*Value*	*Parameter Name*	*Value*
var_smoothing	1.0	C,d, γ	100,1,0.1
		C - Regularization parameter. d - Degree of the kernel function. γ - Kernel coefficient.	

Figure 5 illustrates the PCA technique's analysis. In this study, information on trends is extracted from each technical indicator. Then, to reduce data dimensionality employs a PCA to identify the crucial elements from trend-related information that affect stock prices. When high-weighted principal components are employed to forecast equity market trends, accuracy is significantly improved and the risk associated with trend prediction is significantly reduced. Based on the findings from the PCA method results, the scatter matrix for the first three primary components is displayed in Figure 5(a). These three primary components carry high-weighted principal components. Figure 5(b) displays the number of the principal component vs its associated eigenvalue, ranked from greatest to smallest. The explained variance is displayed here as a function of the primary components using a scree plot.

In Table 5 each machine learning technique was assessed with a different number of PCs (optimal parameter settings and pre-processing methodologies were used) to learn how PCA affects their performance. The performance comparison of the traditional individual classifier and proposed PCA-ML hybrid model on the Nifty 50 Index is shown in Table 5 and it shows that the proposed hybrid model prediction performance outperformed the standard individual classifier utilizing the principal components of high-weighted features based on the experiments done. The experimental findings suggest that the technical factors are signified as trend signals and that the PCA approach combined with ML models outperforms the comparative models in prediction performance.

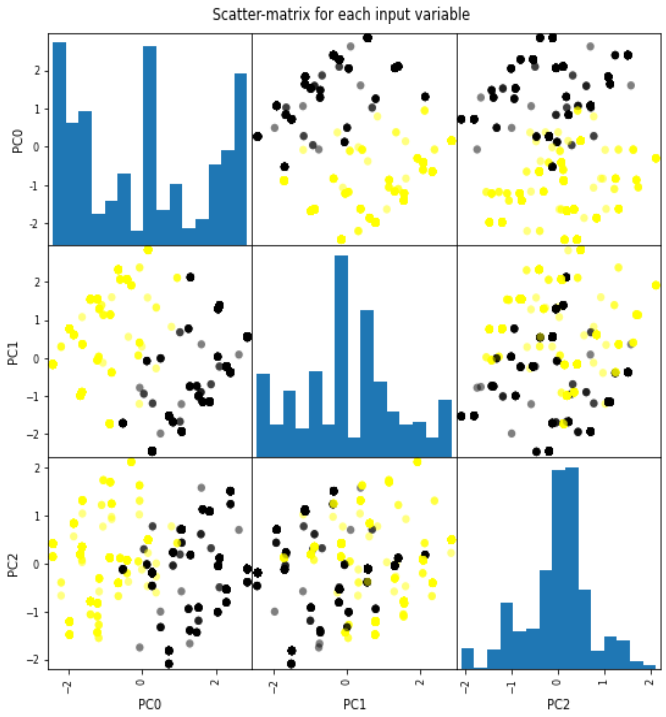
According to the findings of the experiments, the proposed PCA-ML algorithms equally outperformed using a smaller number of PCs compared to the higher number of PCs. Utilizing the first three principal components (percentage of explained variance=80%), experiments on the Nifty 50 index show that SVC (RBF kernel) achieves good accuracy of (0.9968) and F1-score (0.9969), and the random forest (RF) model achieves an accuracy of (0.9969) and F1-Score (0.9968). In AUC performance, SVC (RBF and Linear kernels) and RF have AUC scores of 1.

Table 5. Classification performance comparison of traditional individual classifier and proposed hybrid model on nifty 50 index transformed data

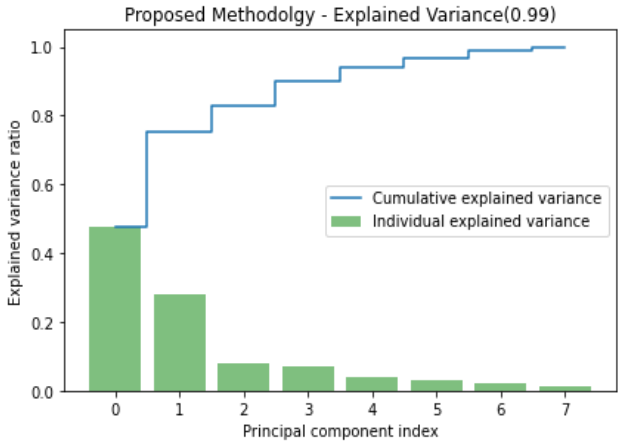
Proposed hybrid methodology (PCA-ML) with trend deterministic layer using technical indicators of Table 1 with different explained variance				
Model	Accuracy	F1-Score	ROC-AUC	Training time (seconds)
PCA Components = 8, Explained Variance=0.99				
PCA+ANN	0.9984	0.9984	0.9999	1.556
PCA+SVC (poly kernel)	0.9953	0.9954	0.9999	0.040
PCA+SVC (RBF kernel)	0.9937	0.9938	0.9999	0.094
PCA+SVC (linear kernel)	0.9953	0.9954	0.9999	0.012
PCA+GNB	0.9968	0.9969	0.9999	0.001
PCA+RF	0.9968	0.9969	0.9984	0.010
PCA Components = 6, Explained Variance=0.95				
PCA+ANN	0.9984	0.9984	0.9999	1.306
PCA+SVC+Poly Kernel	0.9937	0.9938	0.9999	0.018
PCA+SVC+RBF Kernel	0.9984	0.9984	0.9999	0.028
PCA+SVC+Linear Kernel	0.9984	0.9984	0.9999	0.025
PCA+GNB	0.9875	0.9878	0.9961	0.002
PCA+RF	0.9984	0.9984	0.9984	0.026
PCA Components = 5, Explained Variance=0.90				
PCA+ANN	0.9984	0.9984	0.9999	1.455
PCA+SVC+Poly Kernel	0.9937	0.9938	0.9999	0.016
PCA+SVC+RBF Kernel	0.9937	0.9938	0.9998	0.023
PCA+SVC+Linear Kernel	0.9937	0.9938	0.9999	0.032
PCA+GNB	0.9875	0.9878	0.9952	0.001
PCA+RF	0.9984	0.9984	0.9999	0.024
PCA Components = 4, Explained Variance=0.85				
PCA+ANN	0.9969	0.9968	0.9998	1.258
PCA+SVC+Poly Kernel	0.9937	0.9938	0.9995	0.044
PCA+SVC+RBF Kernel	0.9937	0.9938	0.9994	0.097
PCA+SVC+Linear Kernel	0.9813	0.9818	0.9998	0.026
PCA+GNB	0.9797	0.9802	0.9996	0.002
PCA+RF	0.9968	0.9969	0.9984	0.008
PCA Components = 3, Explained Variance=0.80				
PCA+ANN	0.9953	0.9953	0.9983	1.2318
PCA+SVC+Poly Kernel	0.9813	0.9818	0.9996	0.014
PCA+SVC+RBF Kernel	0.9968	0.9969	0.9999	0.019
PCA+SVC+Linear Kernel	0.9937	0.9938	0.9998	0.029
PCA+GNB	0.9797	0.9802	0.9995	0.001
PCA+RF	0.9969	0.9968	0.9999	0.008
Traditional individual classifier without trend deterministic layer using TI of Table 1				
ANN	0.6560	0.6630	0.7199	1.432
SVC+Poly Kernel	0.6672	0.6775	0.7280	0.252
SVC+RBF Kernel	0.6523	0.6804	0.7226	0.248
SVC+Linear Kernel	0.6411	0.6643	0.7187	0.133
GNB	0.6242	0.6655	0.6557	0.002
RF	0.6560	0.6642	0.6922	0.001

The individual classifier performance results using the technical indicators of [19] are shown in Table 6; from this Table 6, it is also clear that our proposed hybrid model its results shown in Table 5 have a significant improvement in the prediction performance compared to technical indicators of [19]. ROC-AUC is a probability curve that displays the TPR against the FPR at different threshold levels, thereby separating the 'signal' from the noise. The area under the curve (AUC) summarizes the ROC curve that measures a classifier's ability to distinguish between classes. When AUC=1, the classifier can successfully differentiate between all positive and negative class points. The ROC curves of the traditional individual classifier using the technical indicators of Table 1 results are depicted in Figure 6(a), and the prediction performance of all the models is poor compared to the proposed hybrid PCA-ML model, and its ROC curves are shown in Figure 6(b).

The ROC curve of the suggested customized PCA-ML using the first three principal components (EV=80%) is shown in Figure 6(b). It can be seen that the performance of ANN, SVM, NB, and RF using the proposed hybrid model has a higher ROC-AUC. SVC (RBF and linear kernels) and RF have AUC scores of 1. These four machine learning classifiers produce curves closer to the top-left corner, indicating more extraordinary performance and lower false-negative and false-positive error rates. It means that all of these ML models are the best for all sensitivity and specificity values.



(a)



(b)

Figure 5. Results of the PCA analysis (a) the scatter matrix of the first three principal components and (b) scree plot of the percentual variability explained by each principal component

Table 6. Classifier performance results

Classifier performance discrete-valued comparison data set using technical features of [19]		
Model	Accuracy	F-measure
ANN	0.8724	0.8770
SVM	0.8909	0.8935
NB	0.8952	0.8990
RF	0.8952	0.8977

These experimental results allow us to draw the following conclusions: i) the purpose of this study was to compare the performance of various supervised machine learning algorithms in predicting the equity market and increasing the predictive potential of hybrid ML algorithms, ii) the PCA approach reduces data dimensionality and enhances model performance by identifying the key features from several technical indicators that influence stock prices, iii) our results show that when deterministic trend data is integrated with PCA-ML models, the performance of the models dramatically increases. The highest accuracy, F1 scores, and shortest training times are attained by all four of the ML approaches utilized in this study, iv) AUC values that are higher suggest a better match. The AUC is greater for all proposed hybrid models' performance, and v) the proposed hybrid methodology might be the best choice for classifying equity market trends using ML algorithms used in this study.

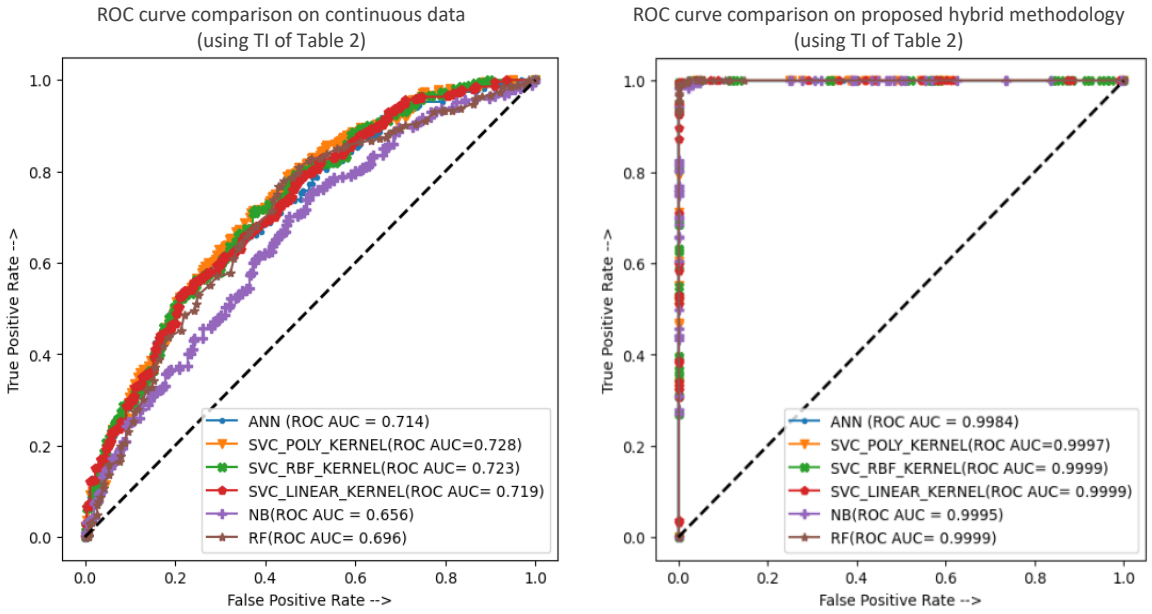


Figure 6. ROC curves of traditional ML classifiers vs. proposed hybrid ML classifiers (a) ROC curves of various traditional ML classifiers without trend deterministic layer using TI of Table 1 and ROC curves of proposed hybrid ML (PCA-ML) classifiers with trend deterministic layer (#PC=3) using TI of Table 1

4. CONCLUSION

The purpose of this study was to apply several machine-learning techniques to forecast the movement of stock prices. In this paper, we classify the Nifty 50 equities market trends using a PCA-ML model hybrid. The suggested model achieves its objectives by first employing feature engineering to generate the financial and technical analysis time series data, and then using the trend deterministic data preparation layer to turn the data into a discrete form (+1 or -1). The principal component analysis (PCA) method is then applied to discrete-form data to introduce the dimensionality of a data set with numerous interrelated qualities while keeping the maximum variability in the data set. The accuracy of the tailored PCA-ML forecasting model is evaluated using classification metrics. A distinct contribution to the research is the introduction of this thorough feature engineering and a customized PCA-ML forecasting model. When discrete/trend data is integrated with PCA-ML models, the performance of the models improves significantly, according to our experiments. Hence, the proposed hybrid approach, which incorporates ANN, SVM, NB, and RF ML algorithms, is possibly the most effective way to classify equity market trends in the stock market.

ACKNOWLEDGEMENTS

We appreciate having the technology resources necessary to do our research at CHRIST (Deemed to be University) Bangalore, India.

REFERENCES

[1] S. Pyo, J. Lee, M. Cha, and H. Jang, "Predictability of machine learning techniques to forecast the trends of market index prices: Hypothesis testing for the Korean stock markets," *PLOS ONE*, vol. 12, no. 11, Nov. 2017, doi: 10.1371/journal.pone.0188107.

[2] G. Box, "Box and Jenkins: time series analysis, forecasting and control," in *A Very British Affair*, London: Palgrave Macmillan UK, 2013, pp. 161–215.

[3] V. Ş. Ediger and S. Akar, "ARIMA forecasting of primary energy demand by fuel in Turkey," *Energy Policy*, vol. 35, no. 3, pp. 1701–1708, Mar. 2007, doi: 10.1016/j.enpol.2006.05.009.

[4] C.-H. Cheng, T.-L. Chen, and L.-Y. Wei, "A hybrid model based on rough sets theory and genetic algorithms for stock price forecasting," *Information Sciences*, vol. 180, no. 9, pp. 1610–1629, May 2010, doi: 10.1016/j.ins.2010.01.014.

[5] G. Ji, J. Yu, K. Hu, J. Xie, and X. Ji, "An adaptive feature selection schema using improved technical indicators for predicting stock price movements," *Expert Systems with Applications*, vol. 200, Aug. 2022, doi: 10.1016/j.eswa.2022.116941.

[6] J. L. Ticknor, "A Bayesian regularized artificial neural network for stock market forecasting," *Expert Systems with Applications*, vol. 40, no. 14, pp. 5501–5506, Oct. 2013, doi: 10.1016/j.eswa.2013.04.013.

[7] Y. Son, D. Noh, and J. Lee, "Forecasting trends of high-frequency KOSPI200 index data using learning classifiers," *Expert Systems with Applications*, vol. 39, no. 14, pp. 11607–11615, Oct. 2012, doi: 10.1016/j.eswa.2012.04.015.

[8] S.-H. Liao and S.-Y. Chou, "Data mining investigation of co-movements on the Taiwan and China stock markets for future investment portfolio," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1542–1554, Apr. 2013, doi: 10.1016/j.eswa.2012.08.075.

[9] H. Park, N. Kim, and J. Lee, "Parametric models and non-parametric machine learning models for predicting option prices: empirical comparison study over KOSPI 200 Index options," *Expert Systems with Applications*, vol. 41, no. 11, pp. 5227–5237, Sep. 2014, doi: 10.1016/j.eswa.2014.01.032.

[10] M.-C. Tsai, C.-H. Cheng, M.-I. Tsai, and H.-Y. Shiu, "Forecasting leading industry stock prices based on a hybrid time-series forecast model," *PLOS ONE*, vol. 13, no. 12, Dec. 2018, doi: 10.1371/journal.pone.0209922.

[11] I. Jolliffe, "Principal Component Analysis," in *International Encyclopedia of Statistical Science*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1094–1096.

[12] H. Ince and T. B. Trafalis, "Kernel principal component analysis and support vector machines for stock price prediction," *IIE Transactions*, vol. 39, no. 6, pp. 629–637, Mar. 2007, doi: 10.1080/07408170600897486.

[13] M. Ghorbani and E. K. P. Chong, "Stock price prediction using principal components," *PLOS ONE*, vol. 15, no. 3, Mar. 2020, doi: 10.1371/journal.pone.0230124.

[14] H. Yu, R. Chen, and G. Zhang, "A SVM stock selection model within PCA," *Procedia Computer Science*, vol. 31, pp. 406–412, 2014, doi: 10.1016/j.procs.2014.05.284.

[15] X. Zhong and D. Enke, "Forecasting daily stock market return using dimensionality reduction," *Expert Systems with Applications*, vol. 67, pp. 126–139, Jan. 2017, doi: 10.1016/j.eswa.2016.09.027.

[16] Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul stock exchange," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5311–5319, May 2011, doi: 10.1016/j.eswa.2010.10.027.

[17] Y. Shynkevich, T. M. McGinnity, S. A. Coleman, A. Belatreche, and Y. Li, "Forecasting price movements using technical indicators: Investigating the impact of varying input window length," *Neurocomputing*, vol. 264, pp. 71–88, Nov. 2017, doi: 10.1016/j.neucom.2016.11.095.

[18] M. Thenmozhi and G. Sarath Chand, "Forecasting stock returns based on information transmission across global markets using support vector machines," *Neural Computing and Applications*, vol. 27, no. 4, pp. 805–824, May 2016, doi: 10.1007/s00521-015-1897-9.

[19] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, "Predicting stock market index using fusion of machine learning techniques," *Expert Systems with Applications*, vol. 42, no. 4, pp. 2162–2172, Mar. 2015, doi: 10.1016/j.eswa.2014.10.031.

[20] M. Göçken, M. Özçalıcı, A. Boru, and A. T. Dosdoğru, "Integrating metaheuristics and artificial neural networks for improved stock price prediction," *Expert Systems with Applications*, vol. 44, pp. 320–331, Feb. 2016, doi: 10.1016/j.eswa.2015.09.029.

[21] 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, doi: 10.1109/ACCESS.2018.2806180.

[22] P.-F. Pai and C.-S. Lin, "A hybrid ARIMA and support vector machines model in stock price forecasting," *Omega*, vol. 33, no. 6, pp. 497–505, Dec. 2005, doi: 10.1016/j.omega.2004.07.024.

[23] D. Kumar, S. S. Meghwani, and M. Thakur, "Proximal support vector machine based hybrid prediction models for trend forecasting in financial markets," *Journal of Computational Science*, vol. 17, pp. 1–13, Nov. 2016, doi: 10.1016/j.jocs.2016.07.006.

[24] NSE Indices, "Historical data reports," niftyindices.com. https://niftyindices.com/reports (accessed Jan. 25, 2023).

[25] C. Manjunath, B. Marimuthu, and B. Ghosh, "Deep learning for stock market index price movement forecasting using improved technical analysis," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 5, pp. 129–141, Oct. 2021, doi: 10.22266/ijies2021.1031.13.

[26] B. Ghosh and E. Kozarević, "Identifying explosive behavioral trace in the CNX Nifty index: a quantum finance approach," *Investment Management and Financial Innovations*, vol. 15, no. 1, pp. 208–223, Mar. 2018, doi: 10.21511/imfi.15(1).2018.18.

[27] B. Ghosh and K. MC, "Econophysical bourse volatility – Global Evidence," *Journal of Central Banking Theory and Practice*, vol. 9, no. 2, pp. 87–107, May 2020, doi: 10.2478/jcbtp-2020-0015.

[28] B. Ghosh, K. M.C., S. Rao, E. Kozarević, and R. K. Pandey, "Predictability and herding of bourse volatility: an econophysics analogue," *Investment Management and Financial Innovations*, vol. 15, no. 2, pp. 317–326, Jun. 2018, doi: 10.21511/imfi.15(2).2018.28.

[29] X. Xu, C. Zhou, and Z. Wang, "Credit scoring algorithm based on link analysis ranking with support vector machine," *Expert Systems with Applications*, vol. 36, no. 2, pp. 2625–2632, Mar. 2009, doi: 10.1016/j.eswa.2008.01.024.

[30] M. Nabipour, P. Nayyeri, H. Jabani, A. Mosavi, E. Salwana, and S. S., "Deep learning for stock market prediction," *Entropy*, vol. 22, no. 8, Jul. 2020, doi: 10.3390/e22080840.

[31] M. Nabipour, P. Nayyeri, H. Jabani, S. S., and A. Mosavi, "Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis," *IEEE Access*, vol. 8, pp. 150199–150212, 2020, doi: 10.1109/ACCESS.2020.3015966.

[32] R. Khemchandani, Jayadeva, and S. Chandra, "Knowledge based proximal support vector machines," *European Journal of Operational Research*, vol. 195, no. 3, pp. 914–923, Jun. 2009, doi: 10.1016/j.ejor.2007.11.023.

BIOGRAPHIES OF AUTHORS

Chinthakunta Manjunath received a B.E. from PESIT, VTU, Bengaluru, in 2007, and M.Tech. from RVCE, VTU, Bengaluru, in the field of Computer Science Engineering in 2011. He is currently an assistant professor in the Department of Computer Science and Engineering, CHRIST (Deemed to be University), Bengaluru. His work focuses on the use of machine learning and deep learning to predict stock market movements in the equity market. He can be contacted at email: manju.chintell@gmail.com.

Balamurugan Marimuthu received his Ph.D. degree in Computer Science from Anna University in Chennai, Tamil Nadu, India. He is currently an associate professor in the Department of Computer Science and Engineering, CHRIST (Deemed to be a university), Bengaluru. Financial market forecasts, wireless networks, and deep learning and machine learning applications are all areas of study that interest him. You can send an email to balamurugan.m@christuniversity.in to get in touch with him. He can be contacted at email: balamurugan.m@christuniversity.in.

Bikramaditya Ghosh holds a Ph.D. in Financial Econometrics from Jain University. He presently holds the position of professor at Symbiosis Institute of Business Management (SIBM), Symbiosis International (Deemed University), in Bengaluru, India. He was an ex-Investment Banker turned applied finance and analytics researcher and practitioner. He has published more than twenty international research papers in finance and economics in journals of repute. He can be contacted at email: bikram77777@gmail.com.