International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V73I3P107
Volume 73 Issue 3, 95-103, March 2025

Original Article

A Comparative Analysis of Various Soft Computing Techniques for Indian Stock Market Prediction

Lavanya Balaji[1], Anita HB[2], Balaji Ashok Kumar[3]

[1,2]*Department of Computer Science, CHRIST (Deemed to be University), Karnataka, India.*
[3]*Department of Commerce, Government First Grade College Vemagal, Karnataka, India.*

[1]*Corresponding Author : lavanya.r@res.christuniversity.in*

Received: 26 September 2024 Revised: 04 January 2025 Accepted: 04 February 2025 Published: 28 March 2025

Abstract - *Soft computing techniques have been increasingly used for stock market analysis in the past few years because they can capture nonlinear aspects which traditional econometric models do not adequately capture. With different techniques like Artificial Neural Networks, Deep Neural Networks and Stacked Autoencoders available, in this paper, the author tries to determine which of the above methods can model the Indian stock market with higher accuracy. In this study, high-frequency data from Nifty 50 is used, and various feature selection techniques such as PCA and linear regression are used for each of the above machine learning models to predict the Nifty 50 data. Finally, all predictions from the different techniques are compared with the actual index movement and the best method for Nifty 50 is suggested.*

Keywords - *Artificial Neural Network, Deep Neural Network, Nifty 50, Stock market, Principal Component Analysis (PCA).*

1. Introduction

Stock movement is highly volatile and is influenced by factors like monetary policy, fiscal policy, innovations, exchange rates, trade agreements in other countries, investor sentiments, etc. Hence, modelling the stock market index movement is challenging [6]. It propounded that stock prices change as soon as new information enters the market. The two strategies historically employed to forecast a company's stock price have been highlighted in most of the literature on stock price prediction. The technical analysis method forecasts the market price of a specific company by examining historical closing and opening price records, trading volume, and closing price trends over the previous several days. The second kind of analysis is categorical and, like the first, depends on outside variables like the business, the market, political and economic aspects, and financial data from news articles, blogs, social media posts, and other economic sources.

In order to forecast stock values, advanced intelligent techniques that rely on technical or fundamental analysis are used today. In particular, when it comes to stock market analysis, a large amount of data with nonlinear patterns must be handled by an efficient model that can recognize the hidden patterns and complex relationships within this large dataset. Machine learning algorithms in this field have shown a significant increase in efficiency, ranging from 60 to 86 per cent compared to traditional methods. Researchers and analysts have used traditional statistical and econometric techniques to predict stock price movement based on historical data like regression and ARIMA. However, less information can be mined from historical data, making predicting the stock market difficult. These traditional statistical techniques primarily use linear aspects of data to predict data, which is a serious setback [4]. With the advent of soft computing techniques like Artificial Neural Networks, Deep Neural Networks, Stacked autoencoder Networks and others, it is now possible to model even nonlinear aspects of data to model the stock market. Machine learning aims to endow computer systems to gain learning skills [10]. Supervised learning is the most frequently used machine learning technique in stock market prediction [22]. A rapidly expanding field of statistical and financial research that reconstructs financial time series data using machine learning techniques to forecast stock market prices and assess performance. Machine learning and profound learning developments have created new opportunities for building models that forecast stock prices from time-series data with high cardinality.

Future changes in stock prices are forecasted using these models. There is a significant research gap in recent research papers, and most researchers have not implemented feature selection or reduction techniques to remove redundant features. The redundant features often cause the problem of overfitting. The prediction value is compromised, and the error metrics values are increased. Hence, there is a significant need to compare the performance of the benchmark models and models with the feature selection technique. This comparative analysis assists the researchers in understanding the performance and the significance of feature selection techniques. In this paper, the researcher has considered neural networks such as ANN, DNN and Stacked Auto Encoder along with feature selection techniques such as linear regression and PCA. The researcher has analyzed several models, and the best model was obtained concerning standard evaluation metrics such as RMSE, MSE, MAE, ME, ACF1, and Theil's U.

2. Literature Review

2.1. Governance and Ethics

Concerns about market manipulation, automated trading, and regulatory monitoring are among the significant ethical and governance issues brought up by the study of stock market prediction using the Nifty 50. Machine learning, High-Frequency Trading (HFT), and algorithmic trading have revolutionized financial markets due to technological advancements, necessitating an analysis of their ethical implications [23]. Market manipulation is one of the main ethical issues where traders or financial institutions manipulate stock prices unscrupulously. Spoofing, in which traders place massive fictitious orders to generate a false feeling of demand and then cancel them, and pump-and-dump schemes, in which stock prices are artificially boosted through deceptive projections before being sold at a profit, are examples of standard manipulative techniques [22].

Furthermore, Artificial Intelligence (AI)-driven forecasts, social media bots, and fake news can all be used to manipulate sentiment and cause needless market volatility. The unequal access to prediction technologies is another significant ethical dilemma. Hedge funds and institutional investors can execute trades faster than regular investors because they frequently have better access to data, processing capacity, and AI models. Businesses may conduct deals in milliseconds thanks to High-Frequency Trading (HFT), which takes advantage of modest price inefficiencies that smaller traders cannot exploit. Concerns regarding the fairness of financial markets are also raised because certain companies obtain insider knowledge and privileged financial data. Regulatory agencies must guarantee transparency and equal access to financial data, and ethical governance should make sure that predictive technologies do not unfairly favor certain firms [23].

From the governance standpoint, the Securities and Exchange Board of India, or SEBI, is essential in controlling algorithmic trading and stopping market manipulation. Pre-approval of trading algorithms, circuit breakers to prevent flash crashes, and real-time market surveillance to identify manipulative trading patterns are just a few of the laws that SEBI has put in place. Governance frameworks must, however, constantly adjust to new technological challenges as AI-based trade develops [23]. Maintaining market stability and shielding individual investors from unethical trading practices require that AI-based trading techniques adhere to regulatory standards. Responsible financial practices are necessary for the ethical application of AI in stock market prediction and regulatory control. The design of AI-driven trading algorithms should steer clear of bias, stop needless market volatility, and prioritize long-term financial stability over immediate financial gain [22]. Regulators should audit financial institutions' usage of AI models to ensure that moral principles are followed.

2.2. Practical Applications of Stock Market Prediction for Nifty 50

The Nifty 50 index's stock market prediction models have many real-world uses, including algorithmic trading, risk management, investment strategy development, and regulatory supervision. To maximize portfolio performance and reduce potential risks, investors use these prediction models to help them make well-informed decisions about purchasing or selling NIFTY 50 stocks [24]. Financial institutions use automated tactics powered by these models to make transactions quickly in algorithmic and high-frequency trading, taking advantage of small market inefficiencies. Risk management specialists use predictive analytics to anticipate market declines or increased volatility, which makes it possible to create efficient hedging plans to safeguard assets.

To maintain market integrity, regulatory agencies like the Securities and Exchange Board of India (SEBI) use these models to track market activity, identify irregularities, and stop manipulative tactics. Retail investors can obtain customized recommendations based on predictive insights thanks to the emergence of AI-driven robo-advisors, which has also democratized access to individualized investment advice. These models' accuracy has been significantly increased by combining machine learning and deep learning techniques, making them essential instruments in today's financial markets. Numerous facets of stock market prediction and its uses have been the subject of recent research [25].

2.3. Feature Engineering

A crucial step in stock market prediction is feature engineering, which involves turning unstructured financial data into valuable inputs that improve predictive models' effectiveness. Analysts can better forecast by capturing complicated market dynamics by deriving algorithmic characteristics like sentiment scores, statistical metrics, trend indicators, momentum oscillators, and volatility measurements. Recent artificial intelligence and machine learning developments have improved these methods, allowing for more complex analysis and use in actual trading situations. Various machine learning algorithms have been employed in stock market prediction to analyze and forecast market movements. The study [20] highlights how well these algorithms process and evaluate extensive data in real time, demonstrating how different machine-learning techniques can be utilized to anticipate stock values. To improve prediction accuracy, the study [21] highlights the value of pre-processing methods to deal with the time series data from the stock market's complicated, dynamic, and nonlinear character and combine several neural network topologies.

2.4. Feature Reduction

A feature selection dimension reduction method is used to choose characteristics pertinent to machine learning tasks. Reducing the amount of the dataset by removing unnecessary and duplicate features is essential for improving machine learning algorithm performance, accelerating learning, and creating basic models. Most feature selection algorithms can work with labelled or unlabeled data, but unified feature selection algorithms can work with both types. Feature extraction, or feature construction, approaches create new features using existing features without losing any information. One well-known example of a feature extraction algorithm is Principal Component Analysis (PCA) [27]. Feature selection is categorized as supervised, unsupervised, and semi-supervised, depending on the availability of labels. The different machine learning techniques use input features to develop a model that predicts a target object.

For a robust model with highly accurate predictability, there is a need to shortlist good and representative features from the list of all available features, which includes relevant, redundant and irrelevant features [9]. This can be done through either Feature selection, Feature extraction or feature generation [12, 17]. Well-known feature selection techniques are Linear regression, Linear correlation, Spearman, and Local learning methods [16, 17]. Pattern recognition can be done by statistical or structural approaches [14].

Different statistical techniques can be used for feature extraction, such as wavelet transformations, Fourier transformations, and Hough transformations. Structural pattern recognitions include Recognition-by-components as propounded in [13]. Variables are ranked using filter methods based on their importance to the current ML algorithms [18]. They serve as a preliminary stage by identifying high-ranking features and utilizing them in machine learning techniques. They disregard the dependence between features but are computationally quick and resist overfitting [8].

2.5. Artificial Neural Network

An Artificial Neural Network is a system that can change its internal structure in response to a function objective. They are inspired by the way the human brain functions. They are especially well-suited to solving nonlinear problems because they can reconstruct the fuzzy rules that determine the best solution for these issues. It is a mathematical structure based on a human neural network, primarily used in pattern recognition, memory, prediction, learning and multivariate analysis. Depending on the calibre and volume of input data, neurons can be arranged topologically (for example, one- or two-dimensional layers, three-dimensional blocks, or more-dimensional structures). The feed-forward topology is how the most popular ANNs are put together. The quantity of input variables determines how many PEs are added to an input layer. One or more hidden layers operating within the ANN receive the information. The result is provided by the output layer, which is the final component of this structure. Whether the output is a single number or a binary value, there is only one PE in the output layer.

A neural network comprises three components: architecture, learning algorithm and activation function [8]. This mathematical structure consists of an input (source) layer, an output (sink) layer and hidden (internal) layers. ANN takes the records at the input layer and produces forecast values at the output layer. ANN will have one input node at the input layer for one variable and can have more than one output node at the output layer. The number of hidden layers can be either equal to one or more than one or zero (no hidden layers) [1].

2.6. Deep Neural Network

Deep Neural Networks (DNN) have shown breakthrough learning capabilities in machine learning due to their deep layers and extremely high parameter dimensions. Because stock-based data is inherently complex, DNNs have demonstrated notable success in learning the relationships and forecasting stock market trends.

In general, a DNN is a combination of input and output layers with multiple hidden layers between them; it is supervised learning to identify the data relationships, which may be linear or nonlinear. Regarding stock price prediction, deep learning algorithms have outperformed conventional approaches. These algorithms demonstrate the ability to process large amounts of data and interpret nonlinear relationships between input variables and prediction goals. Predictive models that use machine learning or linear algorithms do not have this benefit. Deep learning applications in the financial market have been the subject of numerous studies [7, 8, 9]. Among these studies, decision fusion for stock market forecasting is the main focus of the work.

DNN training requires significant processing power, which presents a significant challenge to system design. A model with multiple hidden layers is called a deep neural network. In an ANN model, the hidden layer stores the information regarding each node's importance in the output nodes' forecasted accuracy, along with information on the combination of inputs and their associations. The deep neural network tries to stack additional hidden layers to improve forecasting accuracy [2].

2.7. Stacked Autoencoders Deep Neural Network

Autoencoders are deep neural networks that use unsupervised learning techniques. The autoencoder can learn to compress (encode) and then rebuild (decode) its input. The stacked autoencoder is an unsupervised pretraining that produces hierarchical features by stacking several autoencoders on top of each other, training the encoding portion of each layer to learn progressively abstract representations of the input data [26]. A simple autoencoder is trained on the data first, and then the encoder part (the learnt representation) is used as the input for the subsequent autoencoder in the stack. This process is repeated for each layer in the stack. After training each autoencoder layer separately, supervised techniques (such as backpropagation using labelled data) can fine-tune the entire network. The main advantage of stacked autoencoders is their capacity to efficiently establish deep networks, which improves performance on tasks like regression and classification.

High-dimensional data can be compressed into more manageable formats by stacked autoencoders, which is helpful for tasks like feature extraction, noise reduction, and data visualization. The objective is to learn an adequate representation of the data, typically by encoding that reduces the dimensionality of the input. Like the deep neural network, the stacked autoencoder's deep neural network has three layers: the input, hidden, and output. However, these three layers can be divided into two parts in a stacked autoencoder: deep neural network, unlike deep neural networks, namely, the encoder and the Decoder. The encoder maps the information from the input nodes to the hidden layer, and the Decoder reconstructs the data from the hidden layer to the input nodes [13].

3. Objectives

- Predict the Nifty 50 stock market data using neural networks and feature reduction techniques.
- Extract the best features using the feature reduction technique such as Principal component analysis and linear regression.
- Integrate the feature reduction techniques with the neural networks and build an efficient forecast model.
- Interpret and compare the results using standard evaluation metrics.

4. Methodology

4.1. Dataset

The Nifty 50 data was obtained from the NSE website from 15th October 2007 to 3rd October 2023 and used in this study.

4.2. Research Methodology

Econometric models like ARIMA are usually used to model the stock market. However, in this study, soft computing techniques are used to better capture the nonlinear aspects of historical data. Since time series data is being dealt with first, whether the data is stationary or not is confirmed. This is confirmed through the Augmented Dickey-Fuller (ADF) test and the Phillips-Perron test. If the data is not stationary, it is different. Then, the features of the data can be used for prediction. Then, the features are shortlisted using feature selection and extraction techniques. The predictor feature is selected using the linear regression technique, and feature extraction is done using the principal component technique. The following soft computing techniques are tested namely:

- Artificial neural network with linear regression feature selection technique
- Artificial neural network with principal component analysis feature extraction technique
- Deep neural network with linear regression feature selection technique
- Deep neural network with principal component analysis feature extraction technique
- Stacked Auto Encoder deep neural network.

Finally, all the above techniques are compared, and the best method is suggested.

4.3. Analysis

4.3.1. Stationarity of Data

Data stationarity is essential for precise financial forecasting and stock market prediction. The steady data guarantees consistent patterns for analysis; non-stationary data produces untrustworthy models.

Stock price data can be converted into a stationary format using various methods, including detrending, log transformation, and differencing. Unit Root Tests are used to assess the series' Stationarity [5].

4.3.2 Augmented Dickey-Fuller Test (ADF) at Level

A statistical test called the Augmented Dickey-Fuller (ADF) test assesses if a time series dataset-like stock prices-is stationary or has a unit root, meaning it is not. Since numerous forecasting models are used in stock market prediction, Stationarity is essential. After performing an ADF test using the ‘URCA’ R package [15], failure to reject the Null Hypothesis H0 is obtained, as shown in Table 1, because the gamma of the test statistics is within the” fail to reject” zone.

This suggests that gamma = 0, implying that there is a unit root. However, a0 and a2t are outside the” fail to reject” range, implying that a drift and deterministic trend term exist. As a result, we may deduce that the NSE Nifty 50 series is a non-stationary series with drift and a deterministic trend at the level.

Table 1. ADF test results at FIRST difference Nifty 50 daily

	Value of Test-Statistic	1pct	5pct	10pct
tau3 (γ)	-2.2448	-3.96	-3.41	-3.12
phi2 (a_0)	4.0953	6.09	4.68	4.03
phi3 (a_2t)	4.0771	8.27	6.25	5.34

Fig. 1 Nifty 50 at level

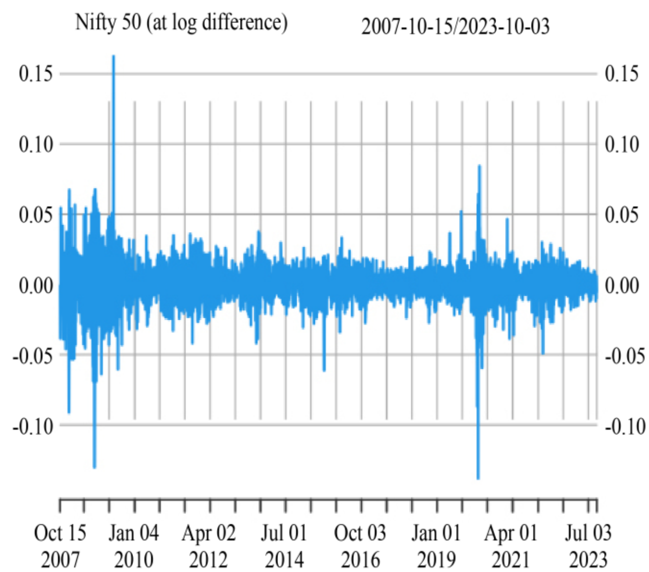

Fig. 2 Nifty 50 (log differenced)

4.3.3. Stationarity Test for First Log Difference Return Series

In order to make the series stationary, we calculate the log difference for the above series. We use log return because:

- Log(1+return) is a normal distribution; thus it fits better for the stochastic pricing model
- Additions of all log(1+return) = Cumulative return of the asset
- Log(1+return) is symmetric as its range is − inf to + ∞.

Table 2. ADF test results at First difference Nifty 50 daily

	Value of Test-Statistic	1pct	5pct	10pct
tau3 (γ)	-43.6282	-2.58	-1.95	-1.62

In Table 2, since γ of the test statistics are outside the" fail to reject" zone, it implies that γ is significant, meaning there is no unit root, so we reject the Null Hypothesis H0.

Hence, we could conclude that the Nifty 50 index return series is a stationary series.

4.3.4. Algorithmic Feature

Several algorithmic features are used in stock market prediction to examine past data, spot trends, and project future price movements. Depending on the methods and data sources, these traits might be divided into several categories. Algorithmic features include both the input (explanatory variable) and the output (explained variable).

Here, the output or the target feature is the Nifty 50 return series (Figure 2). The input or the predictor features are created based on lag values of the past five positions. After creating the predictor features, the data is split into training and testing ranges.

4.3.5. Feature Selection

In this study, feature selection is based on linear regression. The training set of 5 lags is regressed on the Nifty 50 return series, and we get the following output:

Table 3. Nifty 50 training set lags regressed on the training set return values

Coefficients	Estimate	Std. Error	t Value	p Value
(Intercept)	0.0002401	0.0003211	0.748	0.45468
rnse1	0.0687487	0.0212642	3.233	0.00124
rnse2	-0.0209953	0.0212913	-0.986	0.32419
rnse3	-0.0442971	0.0212491	-2.085	0.03722
rnse4	-0.0405284	0.0212595	-1.906	0.05673
rnse5	-0.0331182	0.0212220	-1.561	0.11877

Table 4. Nifty 50 training set lags regressed on the training set return research metrics and its values

Research Metrics	Values
Multiple R-squared	0.01075
Adjusted R-squared	0.008515
F-statistic	4.803 on 5 and 2209 DF
p-value	0.0002269

From the Table 3 and 4, it can be found that only lag 1(rnse1) and lag 3 (rnse3) are statistically significant, and the overall model was also significant (p-value = 0.0002269). The relevant features are selected through regression testing. In the above data set, only lag1 and lag3 were significant.

Hence, instead of modelling all the possible features, only relevant features are used for modelling; thereby, the problem of overfitting is avoided. Based on the above information, the improvement is made concerning regression by only regressing lag 1 (rnse1) and lag 3 (rnse3) on the Nifty 50 return training set, and we have the following output:

Table 5. Nifty 50 training set significant lags are only regressed on training set return values

Coefficient	Estimate	Std. Error	t Value	p-Value
(Intercept)	0.0002215	0.0003214	0.689	0.490674
rnse1	0.0708177	0.0211847	3.343	0.000843
rnse3	-0.0483410	0.0211454	-2.286	-0.022342

Table 6. Nifty 50 training set significant lags are only regressed on the training set return research metrics and its values

Research Metrics	Values
Multiple R-squared	0.007475
Adjusted R-squared	0.006577
F-statistic	8.329 on 2 and 2212 DF
p-value	0.0002489

From the Tables 5 and 6, it can be found that both regressors lag 1(rnse1) and lag 3 (rnse3) are statistically significant, and the overall model was also significant (p-value = 0.0002489).

4.3.6. Feature Extraction

The most important patterns in stock price movements, technical indicators, and fundamental data are captured by Principal Component Analysis (PCA), which is an important dimensionality reduction technique in stock market prediction that reduces the size of high-dimensional data into a minimum set of uncorrelated variables known as principal components. In this paper, principal component analysis is implemented to extract features.

Table 7. Principal component analysis of features

	Standard Deviation	Proportion of Variance	Cumulative Proportion
Comp 1	0.01633382	0.23129049	0.23129049
Comp 2	0.0159175	0.2196505	0.64378546
Comp 3	0.01491463	0.19284447	0.64378546
Comp 4	0.01447546	0.18165488	0.82544035
Comp 5	0.01418995	0.17455965	1.00000000

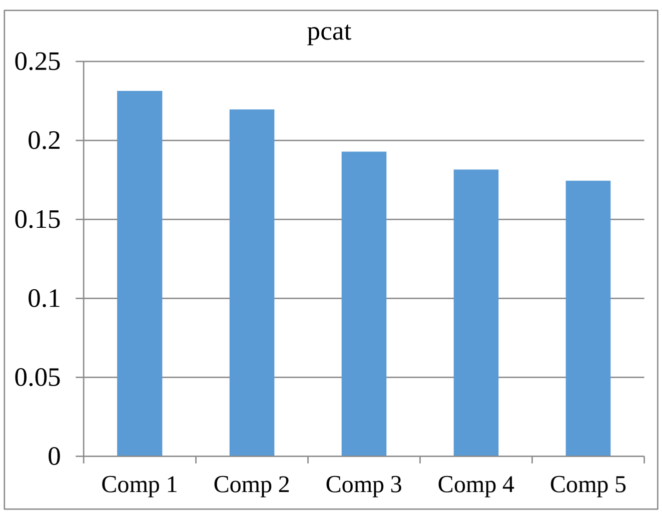

Fig. 3 Feature extraction using PCA for nifty 50 data

Five components were extracted when Principal component analysis was performed with a cumulative frequency of 100%.

4.3.7. Various Models Tested

Artificial Neural Network Based on Feature Selection

In a well-structured artificial neural network, using selected features as inputs, the data is processed through interconnected layers of neurons, learning complex relationships and nonlinear dependencies within stock price movements. This enhances model interpretability and helps prevent overfitting. A well-structured artificial neural network, using selected features as inputs, processes data through interconnected layers of neurons, learning complex relationships and nonlinear dependencies within stock price movements. This enhances model interpretability, helps prevent overfitting, and ensures better generalization of unseen market conditions, making it a powerful tool for financial forecasting.

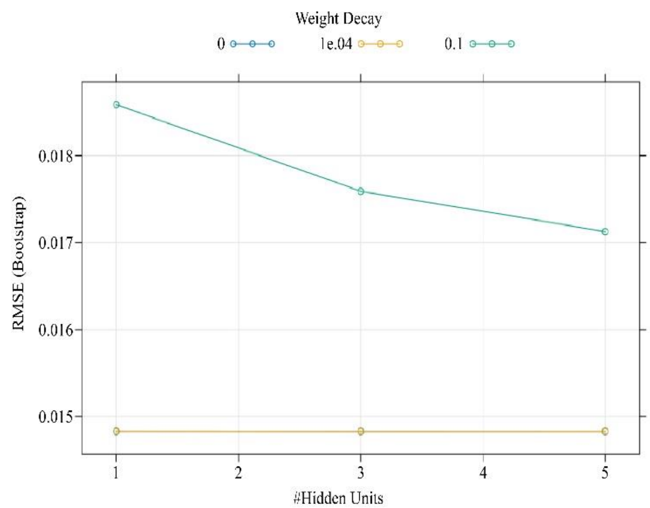

Fig 4. ANN model features selected hidden layer and weight decay

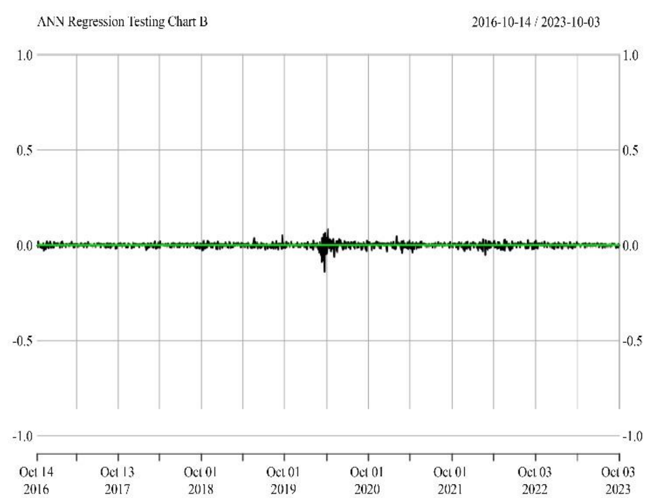

Fig. 5 ANN model features selected hidden layer and weight decay

Artificial Neural Networks (ANNs) combined with feature selection techniques improve stock market prediction by improving model accuracy and reducing computational complexity. The artificial neural network model is implemented in R using the 'R Caret package' (Kuhn and Max, 2008) based on lag 1 and lag 3 (refer to Table 7). The best-tuned model had three hidden units, and its size was 0.01100 (refer to figure 4), for which the RMSE was the lowest. To predict the testing range using the trained model 5 from 14th October 2016. It was found that the returns are within the -0.5 to 0.5 range. In order to find the efficiency of the Artificial neural network model has improved, the" RMSE" values of the accuracy function of the" R forecast package" by (Hyndman et al., 2021) (f["x"] − fitted(f), i.e., actual value less the fitted value) is compared.

The model with the lowest RMSE value is considered the best model. Since the RMSE value of the ANN-PCA feature model is low, we can conclude that it is the best prediction model.

Comparative Metrics of Various Competing Models

From Tables 8 and 9, it can be inferred that the artificial neural network feature selected through PCA is the best of the soft computing techniques employed in this study.

This model accurately models the Nifty 50 stock data compared to other models in this paper. Compared to the other models, the Mean Error (ME), Mean Absolute Error (MAE) and RMSE (Root Mean Squared Error) are lower in the ANN-PCA model.

Table 8. Comparative metrics of different soft computing techniques

Models	ME	RMSE	MSE	MAE
ANN – feature selected through linear regression	3.920129e-05	0.1099232	0.001210783	0.07207152
ANN – feature extracted through PCA	0.000497794	0.01100356	0.0001238358	0.007236843
DNN – feature selected through linear regression	0.0003107702	0.01227211	0.0001506161	0.008025134
DNN – feature extracted through PCA	0.0003272598	0.01226526	0.0001500932	0.008020058
Stacked Autoencoder Deep Neural Network	0.005279825	0.01331772	0.0001510326	0.009433908

Table 9. Comparative metrics using ACF1 and Theil's U for comparing different soft computing techniques

Models	ACF1	Theil's U
ANN – feature selected through linear regression	-0.5319273	0
ANN – feature extracted through PCA	-0.04100471	0
DNN – feature selected through linear regression	-0.052179	0
DNN – feature extracted through PCA	-0.05131984	0
Stacked Autoencoder Deep Neural Network	-0.05329554	0

Theil's U provides insight into the forecasting accuracy of the model. A low Theil's U indicates that the model is predicting close to the actual values, offering a direct metric of prediction performance. Since Theil's U for all the models is zero, all the models are low, i.e. 0. Theil's U directly indicates prediction performance, meaning that the model is making predictions near the actual values.

The correlation between a time series and its lag version is measured using the Auto-Correlation Function (ACF1). Temporal dependencies are common in stock prices, which means that previous prices might impact the current price. ACF1 aids in determining the existence of this link and how well the model has represented it. Temporal dependencies are common in stock prices, which means that previous prices might impact the current price. ACF1 aids in determining the existence of this link and how well the model has represented it. The best model is the ANN with PCA feature selection residuals near zero (refer to Table 9).

5. Conclusion

Prediction of complex time series data like Nifty 50 is a challenging task. Soft computing models offer new methods to model time series data. This study found that artificial neural network features selected through PCA are more accurate than those of other soft computing techniques. We outperformed conventional approaches, such as cutting-edge models like Deep Neural Networks (DNNs) and stacked autoencoders, by combining Artificial Neural Networks (ANNs) with sophisticated feature selection strategies. Our method carefully chooses the most pertinent characteristics, unlike conventional ANNs and DNNs, which frequently experience overfitting due to high-dimensional and noisy data. Predicting the stock market is important as it helps various executives make informed decisions, helps fund managers manage risk, and builds robust portfolios. Using domain-relevant features ensures interpretability and lessens the need for intricate black-box transformations than Stacked Autoencoders, which mostly rely on unsupervised learning to extract features.

Furthermore, although PCA successfully lowers dimensionality, it transforms data into new, uncorrelated components, thus losing significance specific to a particular topic. On the other hand, our method improves model transparency while maintaining feature interpretability. As demonstrated by performance indicators, including lower Mean Squared Error (MSE), we achieved enhanced prediction accuracy, dramatically decreased training time, and increased computing efficiency by optimizing feature selection before ANN training. Outperforming current deep learning and regression-based methods, this combination of neural network flexibility and selective feature extraction produced a strong framework for stock market forecasting. This research gives information about various machine learning techniques to predict the stock market, which the commoner and enterprises can use to understand the stock ma, market trends, which are chaotic and noisy.

5.1. Further Scope of Study

There are several exciting avenues for future study in feature selection-based Artificial Neural Networks (ANNs) for stock market prediction. First, dynamically adjusting to shifting market conditions, including sophisticated feature selection techniques-like hybrid approaches that blend filter, wrapper, and embedding approaches-can improve model performance. Furthermore, combining Deep Reinforcement Learning (DRL) with ANNs can make adaptive feature selection possible. This enables models to determine the most pertinent features in various market situations.

Explainable AI (XAI) techniques to enhance interpretability and increase the transparency of feature selection-driven ANN models for financial analysts and investors represent another investigation line. Complex soft computing models like LSTM and deep belief neural networks can also be compared with previously discussed methods. Hybrid techniques involving econometric models and machine learning models can also be explored. It is, therefore, necessary to update the predictive models regularly during the procedure. Future research should look at how well the results work when the training and testing data are split into sliding window widths of one month, three months, six months, and a year. This is because the movement of stock prices exhibits periodic behaviour over a range of time scales. Future studies could focus on developing major hybrid models that combine feature-selected ANNs with probabilistic models like Bayesian Neural Networks (BNNs) to quantify prediction uncertainty. Last but not least, quantum computing for ANN optimization and feature selection may significantly accelerate computations, resulting in new advancements in high-frequency trading and real-time stock market forecasting.

References

[1] Dinesh Bajracharya, "*Econometric Modeling Vs Artificial Neural Networks: A Sales Forecasting Comparison*," Master's Thesis in Informatics, University of Borås, 2011. [Google Scholar]

[2] Yoshua Bengio, "Learning Deep Architectures for AI," *Foundations and Trends® in Machine Learning*, vol. 2, no. 1, pp. 1-127, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[3] Irving Biederman, "Recognition-by-Components: A Theory of Human Image Understanding," *Psychological Review*, vol. 94, no. 2, pp. 115-147, 1987. [Google Scholar] [Publisher Link]

[4] Yingxuan Chen, Weiwei Lin, and James Z. Wang, "A Dual-Attention-Based Stock Price Trend Prediction Model with Dual Features," *IEEE Access*, vol. 7, pp. 148047-148058, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[5] B.S. Everitt, and A. Skrondal, *The Cambridge Dictionary of Statistics*, Cambridge University Press, pp. 1-168, 2010. [Google Scholar] [Publisher Link]

[6] Eugene F. Fama, "Efficient Capital Markets: A Review of Theory and Empirical Work," *The Journal of Finance*, vol. 25, no. 2, pp. 383-417, 1970. [CrossRef] [Google Scholar] [Publisher Link]

[7] Rob Hyndman et al., "Forecast: Forecasting Functions for Time Series and Linear Models," *CRAN R-Project*, 2021. [CrossRef] [Publisher Link]

[8] Htet Htet Htun, Michael Biehl, and Nicolai Petkov, "Survey of Feature Selection and Extraction Techniques for Stock Market Prediction," *Finance Innovation*, vol. 9, no. 1, pp. 1-25, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] A. Idri, T.M. Khoshgoftaar, and A. Abran, "Can Neural Networks be Easily Interpreted in Software Cost Estimation?," *IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No.02CH37291)*, Honolulu, HI, USA, vol. 2, pp. 1162-1167, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[10] George H. John, Ron Kohavi, and Karl Pfleger, "Irrelevant Features and the Subset Selection Problem," *Machine Learning Proceedings*, pp. 121-129, 1994. [CrossRef] [Google Scholar] [Publisher Link]

[11] Miroslav Kubat, *An Introduction to Machine Learning*, 2nd ed., Springer International Publishing, New York, pp. 1-348, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[12] Max Kuhn, "Building Predictive Models in R Using the caret Package," *Journal of Statistical Software*, vol. 28, no. 5, pp. 1-26, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[13] Mahinda Mailagaha Kumbure et al., "Machine Learning Techniques and Data for Stock Market Forecasting: A Literature Review," *Expert Systems with Applications*, vol. 197, pp. 1-41, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Guifang Liu, Huaiqian Bao, and Baokun Han, "A Stacked Autoencoder-Based Deep Neural Network for Achieving Gearbox Fault Diagnosis," *Mathematical Problems in Engineering*, vol. 2018, pp. 1-10, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[15] Robert Thomas Olszewski, "*Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data*," Master's Theses, Carnegie Mellon University, pp. 1-124, 2001. [Google Scholar] [Publisher Link]

[16] Bernhard Pfaff, *Analysis of Integrated and Cointegrated Time Series with R*, Springer, New York, 2nd ed., pp. 1-190, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[17] Jinwen Sun et al., "Exploiting Intra-Day Patterns for Market Shock Prediction: A Machine Learning Approach," *Expert Systems with Applications*, vol. 127, pp. 272-281, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] Ryan J. Urbanowicz et al, "Relief-Based Feature Selection: Introduction and Review," *Journal of Biomedical Informatics*, vol. 85, pp. 189-203, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[19] Qing-Guo Wang, Xian Li, and Qin Qin, "Feature Selection for Time Series Modelling," *Journal of Intelligent Learning Systems and Applications*, vol. 5, no. 3, pp. 152-164, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[20] Vidushi Tiwari et al., "Stock Market Prediction using Different Machine Learning Algorithms," *10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering*, Gautam Buddha Nagar, India, pp. 147-151, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] Kuldeep Agnihotri et al., "Next-Generation Stock Market Prediction: Integrating CNN and LSTM Model," *2024 International Conference on Data Science and Network Security (ICDSNS)*, Tiptur, India, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[22] Chirag Rathi et al., "Time Series Forecasting for Stock Price Movement: Leveraging NLP for Stock Market Prediction," *Advances in Data and Information Sciences*, vol. 1127, pp. 147-158, Springer, Singapore, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] Odeyemi Olubusola et al., "Machine Learning in Financial Forecasting: A US review: Exploring the Advancements, Challenges, and Implications of AI-Driven Predictions in Financial Markets," *World Journal of Advanced Research and Reviews*, vol. 21, no. 2, pp. 1969-1984, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[24] Pushpendra Singh Sisodia et al., "Stock Market Analysis and Prediction for Nifty50 using LSTM Deep Learning Approach," *2022 2nd International Conference on Innovative Practices in Technology and Management*, Gautam Buddha Nagar, India, pp. 156-161, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Manali Patel, Krupa Jariwala, and Chiranjoy Chattopadhyay, "An Approach toward Stock Market Prediction and Portfolio Optimization in Indian Financial Sectors," *IEEE Transactions on Computational Social Systems*, vol. 12, no. 1, pp. 128-139, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[26] Bo Du et al., "Stacked Convolutional Denoising Auto-Encoders for Feature Representation," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 1017-1027, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[27] Junhua Zheng, Zeyu Yang, and Zhiqiang Ge, "Deep Residual Principal Component Analysis as Feature Engineering for Industrial Data Analytics," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1-10, 2024. [CrossRef] [Google Scholar] [Publisher Link]