International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 11 | Nov 2022 www.irjet.net p-ISSN: 2395-0072

Stock Market Prediction Using Deep Learning

Aayush Shah[1], Mann Doshi[2], Meet Parekh[3], Nirmit Deliwala[4], Prof. Pramila M. Chawan[5]

[1,2,3,4] B.Tech Student, Dept of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India
[5]Associate Professor, Dept of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India

---

Abstract - *The stock market has achieved new heights of popularity in the recent past. The growth in the inflation rate has led people to invest in the stock and commodity markets and other areas of the financial market instead of saving. Technical analysis of stocks with the help of technical indicators has been one of the most popular ways used by traders and investors to help them make decisions like buying or selling in order to make monetary gains. Further, the increase in the availability of computational power and Deep Learning techniques have made it possible to predict the markets to some extent. In this paper, we will make use of various deep-learning algorithms to analyze price trends, predict closing prices and thereby identify trading opportunities. Also we want to leverage our model for predicting and identifying gold prices and prices of various other commodities such as silver , platinum.*

Key Words: Stock Market, Technical Indicators, Deep Learning, Price Trends, Closing Prices

1. INTRODUCTION

Stock market prediction is an extremely ambitious yet essential task for any informed investor. The random, non-linear, non-stationary, and noisy behavior of the market does not help make the task any less challenging. We can program different models to predict the future trends that emerge from observing years and years of stock market data. These trends and patterns help investors make informed decisions to maximize their profits. The erratic nature of the markets could intimidate even the most experienced of investors and therefore it is crucial for the task to be automated to provide some sort of a structure to the ever-fleeting market.

1.1 Problem

Accurately predicting stock market prices is a laborious task. Investors make use of technical and fundamental analysis which in turn make use of certain technical indicators to indicate which stocks to invest in in order to maximize profits for the organization. Even today, a majority of investors and traders manually design and follow a set of rules derived from the said technical & fundamental analysis. This process, however reliable it might be, is tedious and fleeting based on the current market trends. A better and more comprehensive approach should be to use deep learning models to automate the task and make it more generalized in order to encompass a wider range of trends and patterns.

1.2 Complexity

Several models exist for stock market prediction. Selecting the right model that is suitable for our use case is arduous. In addition to this, we also have to choose the right hyperparameters. This involves a lot of trial and error over a large dataset and this process is extremely time-consuming. There might also be times when a certain model is better suited for a certain period of time or for a certain sector. This makes the entire process of stock market prediction severely complex.

1.3 Challenges

The preliminary challenge was to access the publicly available APIs for obtaining accurate and reliable stock market data, one needs to have a trading account, so this was a major inconvenience. The major challenge still lies in the fact that stock indicators are used to accurately predict future trends, we need to calculate the required terms in order to make informed decisions. Another pertinent challenge is the volume of the data involved. In order to make our prediction more accurate, we need to make the dataset more and more inclusive, taking into account several years' worth of stock market data.

2. LITERATURE SURVEY

Stock Market behavior has been studied through various research studies. The research has evolved from using technical indicators to create rules to using deep learning techniques that act as a black box. This black box contains complex rules using which output is given. These rules are learned by the model through the training process.

Logistic regression, Long Short-Term Memory, Support Vector Machine, and Random Forest are some of the most frequent deep learning algorithms that can be used for Stock Market Prediction.

Faraz et al. (2020) have used AE-LSTM for the prediction of the closing price of S&P stocks. The autoencoders are trained to provide a reduced representation of the data by giving importance to appropriate aspects of the data. The reduced representation is then fed to an LSTM-based forecasting network to predict the closing price of the next day. Technical indicators like MACD, RSI, SMA, etc are added as features apart from OHLC and Volume. The authors have performed various things as part of data preprocessing - converting data to stationary time series data, using wavelet transform to denoise the data, removing outliers by the z-score method, and normalizing data using the min-max normalization method. The proposed method performs better than GANs in predicting the next day's closing price as depicted in the paper. After predicting the next day's closing price, buy/sell signal is generated by following a pair of simple rules.

Ghosh et al. (2021) [2] used 2 methods - cuDNNLSTM and Random Forest to predict the probability of the stock giving intraday returns greater than the cross-sectional median intraday returns of all the stocks at time t. Data from the S&P 500 for the period from 1990 to 2018 is considered. The data is broken down by taking windows of size 4 years and stride of 1 year. In each window, approximately 3 years' worth of data is used for training and the rest for testing. 3 types of features are derived for each stock at time t - intraday returns for m days, returns with respect to closing price m days prior, and return with respect to opening price m days prior. The authors have chosen 31 values of m, thus there are a total of 93 features. Random forest uses these 93 features, and LSTM derives its 3 features from 93 features by using Robust Scaler Standardization. Their setting involving LSTM obtains daily returns of 0.64% and that involving Random Forest obtains 0.54% daily returns, both prior to transaction costs.

Tomar et al. (2020) [3] have used various variants of LSTM (Slim Variants) for predicting next-minute price movements (binary classification, 1 for buy and 0 for sell). The variants are then compared using measures such as precision, recall, F1 score, and AUC.

Mabrouk et al. (2022) [4] used a model that stacked CNN Layer, GRU Layer, and Dense Layer upon the Input Layer to predict whether to buy, sell or hold a currency using one-hot encoding. The input layer contains a 2d matrix in which each row contains observation of all features of a particular currency on a specific day. Feature selection techniques were used to reduce the number of features from 150 to less than 30 important input features. Technical indicators are also included in these features.

Paspanthong et al. (2019) [5] experimented with various models with the aim of predicting the next-minute price movements (binary classification, 1 for buy and 0 for sell). Statistically, significant features like SMA, Crossovers, Consecutive Price Trends, etc were selected using Lasso regularization. SPDR S&P 500 Trust (NYSE: SPY) dataset with 1-minute intervals from March 1st until May 24th, 2019. Models like logistic regression, CNN, RNN, LSTM, and SVM were used. SVM that used Polynomial kernels obtained the best accuracy and gave the best returns among other models. The authors highlight that higher accuracy might not always mean greater returns since there can be greater instances of small profits and fewer instances of huge losses. Future scopes as discussed in this paper are - modifying models to take into account the magnitude of profit/loss, assigning weights to different stocks based on their predicted probability, and trading accordingly.

Hamoudi and Elseif (2021) [6] tested two types of models - 3 LSTM and 2 CNN which differed in their architecture. Only the top k (e.g. - 10) percentile of all the trades were labeled as positive trades for training purposes which is consistent with the study of S&P 500 historical returns according to which 10 days in a year are responsible for generating an average of 50% of the total market return of that year and 50 days responsible for about 93%. The dataset is extracted from the Jane Street Market Prediction competition on Kaggle. The authors chose to use a rolling cross-validation approach owing to the sequential nature of the dataset. The models are then trained after replacing missing data points with feature median and normalizing the data. The models are then compared on the basis of measures like precision, recall, and F1 score and using financial performance metrics like the number of trades, Sharpe ratio, Total return, etc. LSTM256x128 was found to be the best among all the other models that were tested with almost double the total returns than any other model.

Miao (2020) [6] employed LSTM to predict the closing price of Amazon, Google, and Facebook after rescaling the data. The author confined the scope to only 3 stocks so that there is no unnecessary influence due to other industries. The author experimented by varying the hyperparameters of the LSTM like layers, dropout, batch size, and epoch, and compared them on the basis of RMSE on the 3 datasets.

3. PROPOSED SYSTEM

3.1 Problem Statement

To implement a model that analyzes the daily prices of all the stocks in an index and predicts trends in the closing

prices of the stocks thus helping in identifying trades that give the most returns. This may help traders/investors to automate the task of shortlisting promising stocks and thus help them boost their profits. First, we obtain OHLC data for all the stocks. Then values of technical indicators and other features are derived from the data. Some preprocessing is then performed on the data after which it is fed to the model as input. The model then predicts the closing price of all the stocks for the next day which helps decide what to buy/sell.

3.2 Problem Elaboration

We aim to build a model that will analyze trends and predict the closing price of all the stocks at time t+1, given their history till time t. Thus, the model helps determine the stocks which will have a high probability of giving good returns.

For the purpose of building such a model, we would require huge amounts of data for training. For now, we will restrict the stocks under consideration to NIFTY 50. We can easily get the daily OHLC data of those stocks from the NSE website. Apart from OHLC, multiple features are calculated using this data.

This data is then used to train the model. After the model is trained, it will be able to help shortlist stocks that can be traded to earn profits by analyzing price trends.

3.3 Proposed Methodology

In order to predict stock prices, we need historical data on all the stocks in NIFTY 50. The data available contains daily OHLC prices and the volume of the stocks. Other features are derived from the price data and appended to the dataset. The Lasso regularization method was used to select statistically significant features by reducing their corresponding coefficients to zero.

Together, the OHLC and the derived features of all the stocks on all days in NIFTY 50 comprises our raw dataset. The NIFTY50 index is updated every 6 months. To avoid survivorship bias this fact is taken into consideration.

The data is then annotated. The intraday gain of every stock is calculated and the trade that occurs in the top k percentile is labeled as 1, the rest are labeled as 0. Rather than trading in every opportunity that is predicted as a probable profitable trade, the model will wait for a trade that is identified to be in the top percentile. This method is consistent with studies of historical returns on the S&P 500 and other market indices showing that the best 10 days in any given year are responsible for generating approximately 50% of the total market return for that year. Furthermore, the best 50 days in any given year are responsible for about 93% of the total return for the whole year. Thus, it is better to focus on identifying the most profitable trading opportunity and avoiding taking an unnecessary risk by acting on every possible trade signal.

After labeling, various pre-processing tasks are performed on the data before feeding the data to the model for training. Random validation and test sets are not appropriate owing to the sequential nature of the data. Hence, a rolling cross-validation approach is adopted.

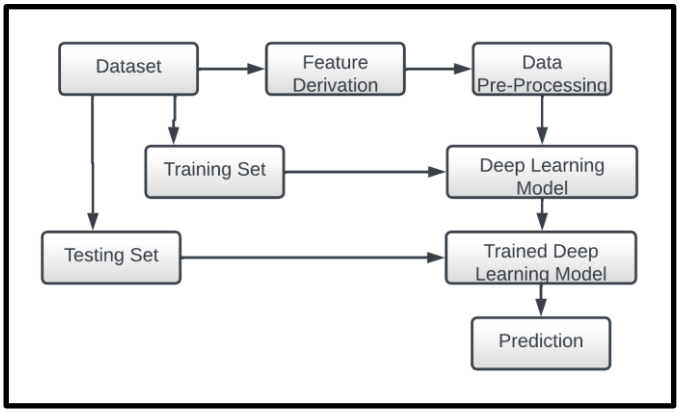


Figure-1: General Methodology of Stock Market Prediction

Data Preprocessing:

Transformation to achieve stationarity: Stationary data refers to the time series data in which the mean and variance do not vary with time. The data is considered non-stationary if a strong trend or seasonality is observed. A common assumption in many time series techniques is that the data are stationary because many statistical analysis models are built upon the assumption that mean and variance are consistent over time.

Noise reduction using Wavelet Transformation: The basic idea behind wavelet denoising, or wavelet thresholding, is that the wavelet transforms lead to a sparse representation for many real-world signals, i.e. the wavelet transform concentrates the signal in a few large-magnitude wavelet coefficients. Wavelet coefficients that are small in value are typically noise, hence they can be reduced or removed without affecting the signal quality. The data is then reconstructed using the remaining coefficients using the inverse wavelet transform.

Outlier removal using z-score: Z score, also called the standard score, is an important concept in statistics that helps detect outliers. Z score value indicated the number of standard deviations a point is from the mean. If the z

score of a data point is more than 3, it indicates an anomaly. Such a data point can be considered an outlier.

Data Normalization: Normalization is used to ensure similar distribution of values across all features. Normalization gives equal importance to each variable, thus preventing a single variable with large values from steering the model in its direction. We have used min-max normalization for normalizing the data since it ensures that the values of all the features lie in the same range of values.

Adding Features:

Technical indicators are heuristic or pattern-based signals that are produced by the price, volume, and/or open interest of a security or contract. They are used by traders who follow technical analysis. Following are some of the technical indicators derived from the closing price of the underlying security to help train the model.

Factors affecting Gold Prices:

Global uncertainty in terms of political and geopolitical aspects is another crucial factor that affects stock prices and gold. The monetary policies adopted by major global Central Banks like the US Fed, the ECB, and the Bank of Japan have a pressing impact on gold prices. Economic growth in key economies like tax cuts, and big pushes to infra have an impact on gold prices as when a country is growing well, investors would typically prefer to participate in this growth through equities and rather not park their monies in gold.

Measuring and leveraging Exchange rates, economic wellness, and central gold reserves of major importers and exporters of gold i.e. Switzerland, U.K., Singapore, Hong Kong, India, China, and Thailand will help us make better predictions of gold and related commodities.

Relative Strength Index:

Relative strength index (RSI) is a technical indicator and displays momentum oscillations. It indicates overbought and oversold conditions for a particular security by measuring the momentum of the respective security.

$$RSI = 100 - \left[\frac{100}{1 + \frac{n_{up}}{n_{down}}}\right]$$

where:
n_{up} = average of n-day up closes
n_{down} = average of n-day down closes
(most analysts use 9 - 15 day RSI)

Moving Average Convergence/Divergence:

Moving average convergence divergence(MACD) is a technical indicator that displays the occurrences of trends based on the momentum of the underlying security. It is derived from two exponential moving averages of the particular security.

MACD line: 12-Period EMA – 26-Period EMA

Signal line: 9-Period EMA

Histogram: Difference between MACD line and signal line

Stochastic:

Stochastic is a momentum-based oscillator of a particular security across the range of its prices over a period of time. It generates indications of overbought and oversold levels of security similar to RSI.

Formula for the Stochastic Oscillator

$$\%K = \left(\frac{C - L14}{H14 - L14}\right) \times 100$$

where:
C = The most recent closing price
L14 = The lowest price traded of the 14 previous trading sessions
H14 = The highest price traded during the same 14-day period
%K = The current value of the stochastic indicator

Bollinger Bands:

Bollinger Bands are technical indicators based on the volatility of the underlying security's price. These bands are plotted at a particular value of the standard deviation of the price of security above and below the actual value.

$$Upper\ Band = Moving\ Average + Constant\sqrt{\frac{\sum_{i=1}^{n}(y_i - Moving\ Average)^2}{n}}$$

$$Lower\ Band = Moving\ Average - Constant\sqrt{\frac{\sum_{i=1}^{n}(y_i - Moving\ Average)^2}{n}}$$

Simple Moving Average:

A simple moving average (SMA) is calculated by taking the average of a selected closing price ranges divided by the number of periods in that range.

$$\text{SMA} = \frac{A_1 + A_2 + \ldots + A_n}{n}$$

where:

A_n = the price of an asset at period n

n = the number of total periods

Exponential Moving Average:

Exponential moving average (EMA) calculates a weighted average of the selected range of price of a security from the number of periods in that range. It gives more weight to the more recent price of the security.

$$EMA_{\text{Today}} = \left(\text{Value}_{\text{Today}} * \left(\frac{\text{Smoothing}}{1 + \text{Days}}\right)\right) + EMA_{\text{Yesterday}} * \left(1 - \left(\frac{\text{Smoothing}}{1 + \text{Days}}\right)\right)$$

where:

EMA = Exponential moving average

Directional Movement Index:

The Directional Movement Index (DMI) is a trend based oscillation indicator that measures the strength of a trend of the security.

$$+\text{DI} = \left(\frac{\text{Smoothed } +\text{DM}}{\text{ATR}}\right) \times 100$$

$$\text{-DI} = \left(\frac{\text{Smoothed -DM}}{\text{ATR}}\right) \times 100$$

$$\text{DX} = \left(\frac{|\ +\text{DI} - \text{-DI}\ |}{|\ +\text{DI} + \text{-DI}\ |}\right) \times 100$$

where:

+DM (Directional Movement) = Current High − PH

PH = Previous high

-DM = Previous Low − Current Low

Smoothed +/-DM = $\sum_{t=1}^{14} \text{DM} - \left(\frac{\sum_{t=1}^{14} \text{DM}}{14}\right) + \text{CDM}$

CDM = Current DM

ATR = Average True Range

3.4 Algorithms

Logistic Regression:

Logistic regression is a process of modeling the probability of an outcome for an input variable. The most common logistic regression models an output that could have values such as true/false, yes/no, and so on. While in Linear Regression MSE or RMSE is used as the loss function, logistic regression makes use of a loss function called maximum likelihood estimation (MLE) which is a conditional probability. If the probability is greater than 0.5, the predictions will be classified into the class 0. Otherwise, class 1 will be assigned.

Convolutional Neural Network (CNN):

Convolutional Neural Network is a feed-forward neural network. Like the classic architecture of a neural network including input layers, hidden layers, and output layers, the convolutional neural network also contains these features and the input of the convolution layer is the output of the previous convolution layer or pooling. The number of hidden layers in a convolutional neural network is more than that in a traditional neural network, which, to some extent, shows the capability of the neural network.

An ordinary CNN model consists of three primary layers: The Convolutional Layer, The Pooling Layer, and The Fully Connected Layer.

Support Vector Machines (SVM):

Support vector machines (SVMs) are a group of supervised learning methods aimed at solving problems related to classification, regression, and outlier detection. SVM's target is to create the best line or decision boundary that can segregate n-dimensional space into classes which enables us to classify a new data point in the correct category in the future. This best decision boundary can be called a hyperplane. SVM picks the extreme points/vectors that help in creating the hyperplane. These boundary cases are called support vectors.

Long Short-Term Memory (LSTM):

Long Short-Term Memory Network is an advanced RNN, a sequential network, that allows information to persist, thus enabling it to make predictions in the time domain. LSTM takes care of the vanishing gradient problem which was an issue in RNN. Since LSTMs are effective at capturing long-term temporal dependencies without suffering from problems faced by RNNs, they have been used in many state-of-the-art models.

The following are the components of an LSTM network:

1. Forget Gate: The forget gate forgets information that is not useful, thus giving importance to what really matters.

2. Learn Gate: Current input and short-term memory are combined together so that necessary information that

we have recently learned can be applied to the current input.

3. Remember Gate: An updated LTM is formed in the Remember gate by combining information obtained from LTM and STM.

4. Use Gate: This gate also uses LTM, and STM as a combined STM to predict the output of the current event

LSTM Autoencoder (AE-LSTM):

AE-LSTM is an implementation of an autoencoder using an Encoder-Decoder LSTM architecture. An encoder-decoder LSTM is configured to encode, decode, and recreate an input sequence.

The performance of the model is evaluated on the basis of to what extent the model recreates the original sequence. After training the encode-decoder model, the decoder part of the model is discarded, leaving just the encoder model. The resulting model is then used to encode input sequences to a fixed-length vector. The resulting vectors can then be used in applications like compression. The training objective forces the encoder to include only useful features/trends in the fixed-length vector, which then serves as an improved input to the prediction model.

Random Forest:

Random forest is a Supervised Machine Learning Algorithm that is used mainly in classification or regression models. It builds decision trees on different samples and performs aggregation on the output given by these decision trees.

Steps involved in the random forest algorithm:

Step 1: First, n number of random records out of k records from the data set.

Step 2: Different decision trees are constructed for each sample.

Step 3: Each decision tree, created on different samples, will generate an output. Thus, we will have multiple intermediate outputs.

Step 4: Majority Voting or Averaging for Classification and regression respectively is performed to compute the final output.

Ensemble Learning:

Ensemble learning is how multiple models, like classifiers or experts, are critically generated and combined to solve problems. Ensemble learning is primarily used to improve the (classification, prediction, function approximation, etc.) performance of a model, or reduce the chances of an unfortunate selection of a less thorough one.

Bagging

By using bootstrapped replicas of the training data the diversity of classifiers in bagging is obtained. From the entire training dataset,with replacement different training data subsets are randomly drawn. Each training data subset is utilized to train a different classifier of the same type. By taking a simple majority vote on their decisions , individual classifiers are then combined . The class chosen by the most number of classifiers is the ensemble decision for any given instance.

Boosting

Even in bagging , we create many classifiers by resampling the data , which further are combined by majority voting . But it differs in resampling of training data .Here we tend to provide most informative training data for each successive classifier .

Explaining further , each iteration of boosting creates 3 weak classifiers , the 1st classifier C1 being trained on a random subset of available training data ,c2 being fed the most informative training data with 50% of which was correctly classified by its predecessor and remaining half misclassified . C3 is trained with instances where C1 and C2 diaggree and all 3 classifiers are combined via 3 way majority .

Reinforcement Learning:

In the context of reinforcement learning, an agent obtains information from the environment. The agent adapts the received data to the environment's current state. The AI then decides on an action to be taken based on the rewards associated with each choice. Moreover, each action alters the environment and the total number of reward points. Reinforcements for rewarding or punishing specific actions are immediately added on the basis of the new state. This interaction between action and environment will persist until the agent masters the art of choosing a decision strategy that maximizes the total return.

Input: Initial conditions for the model to begin with

Output: Possible outputs in accordance with the variety of solutions possible to a particular problem

Training: Based on the current conditions and possible actions and long term goals , the model takes action . And based on its action the model will be punished or rewarded.

The model continues to learn.

The best solution is determined based on the maximum reward.

Few basic elements of an RL problem:

Environment — The physical world in which the agent operates and will receive feedback from .

State — Explains the current situation of the agent

Reward — Feedback received from the environment based on the action

Policy — This helps in choosing which actions to take based on one's current state.

Value — Long term reward that the model aims to achieve

4. CONCLUSION

In this research paper we have provided an overview of the relevant research that has been carried out in similar fields. We provide a review and relative study of different deep learning models that have been used for the purpose of predicting stock market prices. The purpose of a stock market price prediction system is to analyze current trends and predict prices. This will help shortlist probable winning trades thereby providing profits.

REFERENCES

[1] M. Faraz, H. Khaloozadeh and M. Abbasi, "Stock Market Prediction-by-Prediction Based on Autoencoder Long Short-Term Memory Networks," 2020 28th Iranian Conference on Electrical Engineering (ICEE), 2020, pp. 1-5, doi: 10.1109/ICEE50131.2020.9261055.

[2] Ghosh, P., Neufeld, A., & Sahoo, J. K. (2020). Forecasting directional movements of stock prices for intraday trading using LSTM and random forests. arXiv. https://doi.org/10.48550/arXiv.2004.10178

[3] G. Taroon, A. Tomar, C. Manjunath, M. Balamurugan, B. Ghosh and A. V. N. Krishna, "Employing Deep Learning In Intraday Stock Trading," 2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), 2020, pp. 209-214, doi: 10.1109/ICRCICN50933.2020.9296174.

[4] Nabil MABROUK, Marouane CHIHAB, Zakaria HACHKAR and Younes CHIHAB, "Intraday Trading Strategy based on Gated Recurrent Unit and Convolutional Neural Network: Forecasting Daily Price Direction" International Journal of Advanced Computer Science and Applications(IJACSA), 13(3), 2022. http://dx.doi.org/10.14569/IJACSA.2022.0130369

[5] Art Paspanthong, Nick Tantivasadakarn and Will Vithayapalert. (2019). *Machine Learning in Intraday Stock Trading*. Computer Science Department, Stanford University.

[6] Hamdy Hamoudi and Mohamed A Elseif. (2021). *Stock Market Prediction using CNN and LSTM*. Computer Science Department, Stanford University.

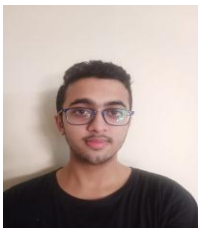
[7] Yan Miao. (2020). *A Deep Learning Approach for Stock Market Prediction*. Computer Science Department, Stanford University.

[8] https://www.motilaloswal.com/blog-details/What-factors-impact-the-price-of-gold-in-the-global-markets/1310

[9] https://www.bankbazaar.com/gold-rate/top-5-factors-that-affect-gold-rate-in-india.html

BIOGRAPHIES

Aayush Shah, B. Tech Student, Dept. of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India.



Mann Doshi, B. Tech Student, Dept. of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India

Meet Parekh, B. Tech Student, Dept. of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India.

Nirmit Deliwala, B. Tech Student, Dept. of Computer Engineering and IT, VJTI College, Mumbai, Maharashtra, India.

Prof. Pramila M. Chawan is working as an Associate Professor in the Computer Engineering Department of VJTI, Mumbai. She has done her B.E. (Computer Engineering) and M.E. (Computer Engineering) from VJTI College of Engineering, Mumbai University. She has 28 years of teaching experience and has guided 80+ M. Tech. projects and 100+ B. Tech. projects. She has published 134 papers in International Journals and 20 papers in National/International Conferences/ Symposiums. She has worked as an Organizing Committee member for 21 International Conferences and 5 AICTE/MHRD sponsored Workshops/STTPs/FDPs. She has participated in 14 National/International Conferences. She has worked as NBA Coordinator of the Computer Engineering Department of VJTI for 5 years. She had written a proposal under TEQIP-I in June 2004 for 'Creating Central Computing Facility at VJTI'. Rs. Eight Crores were sanctioned by the World Bank under TEQIP-I on this proposal. The Central Computing Facility was set up at VJTI through this fund which has played a key role in improving the teaching-learning process at VJTI.