

A Decision Tree- Rough Set Hybrid System for Stock Market Trend Prediction

Binoy.B.Nair

Department Electronics and
Communication Engineering, Amrita
Vishwa Vidayapeetham, Ettimadai,
Coimbatore, 641105, India.

V.P Mohandas

Department Electronics and
Communication Engineering, Amrita
Vishwa Vidayapeetham, Ettimadai,
Coimbatore, 641105, India.

N. R. Sakthivel

Department of Mechanical
Engineering, Amrita Vishwa
Vidayapeetham, Ettimadai, Coimbatore,
641105, India.

ABSTRACT

Prediction of stock market trends has been an area of great interest both to those who wish to profit by trading stocks in the stock market and for researchers attempting to uncover the information hidden in the stock market data. Applications of data mining techniques for stock market prediction, is an area of research which has been receiving a lot of attention recently. This work presents the design and performance evaluation of a hybrid decision tree- rough set based system for predicting the next days' trend in the Bombay Stock Exchange (BSE-SENSEX). Technical indicators are used in the present study to extract features from the historical SENSEX data. C4.5 decision tree is then used to select the relevant features and a rough set based system is then used to induce rules from the extracted features. Performance of the hybrid rough set based system is compared to that of an artificial neural network based trend prediction system and a naive bayes based trend predictor. It is observed from the results that the proposed system outperforms both the neural network based system and the naive bayes based trend prediction system.

General Terms

Data mining, stock market, trend prediction

Keywords

Stock market, Rough set, decision tree, artificial neural networks, Technical indicators, Rules.

1. INTRODUCTION

Stock market prediction has been an area of intense interest due to the potential of obtaining a very high return on the invested money in a very short time. However, according to the efficient market hypothesis[1], all such attempts at prediction are futile as all the information that could affect the behaviour of stock price or the market index must have been already incorporated into the current market quotation. There have been several studies, for example, in [2], which question the efficient market hypothesis by showing that it is, in fact, possible to predict, with some degree of accuracy, the future behaviour of the stock markets. Technical analysis has been used since a very long time for predicting the future behaviour of the stock price. A total of 21 different technical indicators are used in the present study to extract information from the financial time series and hence, they act as the features that are used for stock market trend prediction. A C4.5 decision tree is used to select the relevant features (technical indicators) from the extracted feature set. The selected features are then applied to a rough set based system for predicting one-day-ahead trends in the stock market. An artificial neural network based

trend predictor and a naive bayes based trend prediction system are also implemented. The performance of the rough set based system is compared to that of the artificial neural network based and the naive bayes based predictors. The financial time series considered for the purpose is the Bombay Stock Exchange sensitive index (BSE-SENSEX). The period under consideration is from September 3, 2003 to March 7,2010.

The rest of the paper is organised as follows: Section 2 presents the literature survey, section 3 presents the design of the proposed hybrid system, the comparison of the proposed system with an artificial neural network and a naive bayes based system is presented in section 4 and conclusions are presented in section 5.

2. LITERATURE SURVEY

Artificial neural networks have been widely used prediction of financial time series. In [3], the effectiveness of time delay, recurrent and probabilistic neural networks for prediction of stock trends based on historical data of the daily closing price was attempted. In [4] technical indicators were used as inputs to a feedforward neural network for prediction of Taiwan Stock Exchange and NASDAQ. In [5], technical indicators and a backpropagation neural network was used to create a decision support system for exchange traded funds trading. Technical indicators and neural networks were used in [6] to predict the US Dollar Vs British Pound exchange rates. In [7] a framework for intelligent interaction of automatic trading algorithms with the user was presented. In [8] a back propagation neural network was employed to predict the buy/sell points for a stock and then applied a case based dynamic window to further improve the forecast accuracy. In [2] a survey of more than hundred articles which used neural networks and neuro-fuzzy models for predicting stock markets was presented. It was observed that soft computing techniques outperform conventional models in most cases. Defining the structure of the model is however, a major issue and is a matter of trial and error. In [9], review of data mining applications in stock markets was presented. [10], used a two-layer bias decision tree with technical indicators feature to create a decision rule that can make recommendations when to buy a stock and when not to buy it. [11] combined the filter rule and the decision tree technique for stock trading. [12] presented a hybrid model for stock market forecasting and portfolio selection. In[13] a hybrid fuzzy time series model with cumulative probability distribution approach and rough set rule induction was proposed, to forecast stock markets. Cumulative probability distribution approach was used to discretize the observations in training datasets based on the characteristics of data distribution and rules were generated using rough set

algorithm. Forecasting was then done based on rule support values from rough set algorithm.

3. DESIGN OF THE HYBRID SYSTEM

The hybrid trend prediction system proposed in this paper works in the following way: First the features are extracted from the daily stock market data. Then the relevant features are selected using decision tree. A rough set based classifier is then used to predict the next day's trend using the selected features. In the present study, the prediction accuracy of the proposed hybrid system is validated using the Bombay Stock Exchange Sensitive Index (BSE-SENSEX or commonly, SENSEX) data. The daily SENSEX data (open, high, low, close and volume) from September 3, 2003 to March 7, 2010 (a total of 1625 BSE working days) is considered for the purpose. The performance of trend prediction systems are evaluated using the 10-folds cross validation method.

3.1 Bombay Stock Exchange Sensitive Index (BSE-SENSEX)

The BSE-SENSEX was selected for empirical validation of the proposed trend prediction system. BSE is the world's largest exchange in terms of the number of listed companies (over 4900). It is in the top ten of global exchanges in terms of the market capitalization of its listed companies (as of December 31, 2009). The companies listed on BSE command a total market capitalization of 1.36 trillion US Dollars as of 31st March, 2010. The BSE Index, SENSEX, is India's first and most popular Stock Market benchmark index. The stocks that constitute the BSE-SENSEX are drawn from almost all the market sectors and hence the BSE-SENSEX is capable of accurately tracking the movements of the market. Hence, BSE-SENSEX is taken as the representative of the Indian stock markets. Being such a large and complex system, prediction of trends in the BSE-SENSEX is especially challenging.

3.2 Feature Extraction

The inputs of the proposed system are the daily open, high, low, close, trading volumes of the SENSEX and technical indexes. Twenty one commonly used technical indicators (comprising of twenty five indexes) are considered initially. The Technical indexes considered, are presented in Table 1. Three types of indicators, namely, volume based, price based and overlay indicators were chosen. Since the proposed system is capable of choosing the relevant features automatically, the exact number of indicators is not of much importance, only those indicators which contain any relevant information will be selected by the system. Literature on the technical indices considered above, is widely available, for example in [14]. Classification of the trend into up, down and no trend was done according to the definition in [15] in the following way:

The market is formally classified as being in an uptrend (downtrend) when all the following conditions are satisfied:

1. The closing value must lead (lag) its 25 day moving average
2. The 25 day moving average must lead (lag) 65 day moving average.
3. The 25 day moving average must have been rising (falling) for at least 5 days.
4. The 65 day moving average must have been rising (falling) for at least 1 day.

Table 1. Technical Indicators for feature extraction

Indicator	Type		
Positive volume index	Volume-based indicators		
Negative volume index			
On balance volume			
Price volume trend			
Relative strength Indicator (RSI)	Price-based indicators		
Highest high			
Lowest low			
Moving Average		Nine period moving average	
Convergence			
Divergence consisting of two indexes		MACD line	
Momentum			
Acceleration			
stochastic oscillator consisting of two indexes			%K
			%D
Chaikin's volatility			
William's %R			
Typical price			
price rate of change			
Median price			
Weighted close			
William's accumulation/ distribution			
Bollinger bands consisting of three indexes	Bollinger Upper		
	Bollinger Mid		
	Bollinger Lower		
25-day moving average	Overlays		
65-day moving average			

If the movement of the market cannot be classified as either an uptrend or a downtrend, it is assumed that there is no trend in the market movement. The above information gives the present day's trend and is also used to predict the next day's trend. The aim of all the trend prediction techniques considered in the present study, is to identify the relationship between the information variables and the decision variable. All the technical indexes considered, may or may not be necessary to arrive at the decision. It is also possible that some features may have a negative impact on the accuracy of the system. Hence, feature selection is used to identify those features (technical indexes) which are necessary for trend prediction and help in maintaining or improving the prediction accuracy. Features which are not selected are deemed unnecessary and are discarded.

3.3 Decision Tree based Feature Selection

The Decision tree C4.5 algorithm[15] is widely used to construct decision trees, and is used in the present study to reduce the number of features required for rule generation. A decision tree is a flowchart-like tree structure, where each *internal node* (non-leaf node) denotes a test on an attribute, each branch represents an outcome of the test, and each *leaf node* (or terminal node) holds a class label. The topmost node in a tree is the root node. Given a tuple, for which the associated class label is unknown, the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. The construction of decision tree classifiers does not require any domain knowledge or parameter setting, and

therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high dimensional data. During tree construction, attribute selection measures are used to select the attribute that best partitions the tuples into distinct classes. When decision trees are built, many of the branches may reflect noise or outliers in the training data. Tree pruning attempts to identify and remove such branches, with the goal of improving classification accuracy on unseen data.

3.3.1 Attribute selection measure for decision tree- the Gain Ratio

An attribute selection measure (or a splitting rule) is a heuristic for selecting the splitting criterion that best separates a given data partition, D , of class-labeled training tuples into individual classes. It provides a ranking for each attribute describing the given training tuples. The attribute having the best score for the measure is chosen as the *splitting attribute* for the given tuples. The tree node created for partition D is labelled with the splitting criterion, branches are grown for each outcome of the criterion, and the tuples are partitioned accordingly.

C4.5, uses *gain ratio* as its attribute selection measure. It is defined as:

$$\text{Gain ratio}(A) = \frac{\text{Gain}(A)}{\text{SplitInfo}_A(D)} \quad (1)$$

Where:

A is the attribute under consideration.

$$\text{SplitInfo}_A(D) = - \sum_{j=1}^v \frac{|D_j|}{|D|} \times \log_2 \left(\frac{|D_j|}{|D|} \right) \quad (2)$$

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(D) \quad (3)$$

$$\text{Info}_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times \text{Info}(D_j) \quad (4)$$

$$\text{and } \text{Info}(D) = - \sum_{i=1}^m p_i \log_2 p_i \quad (5)$$

here p_i is the probability that an arbitrary tuple in D belongs to class C_i and is estimated by $|C_{i,D}| / |D|$.

The $\text{SplitInfo}_A(D)$ represents the potential information generated by splitting the training data set, D , into v partitions, corresponding to the v outcomes of a test on attribute A . The attribute with the maximum gain ratio is selected as the splitting attribute.

The $\text{Info}(D)$, also known as the entropy of D , is the average amount of information needed to identify the class label of a tuple in D .

$\text{Info}_A(D)$ is the expected information required to classify a tuple from D based on the partitioning by A . The smaller the expected information still required, the greater the purity of the partitions.

$\text{Gain}(A)$, i.e. the information gain, is defined as the difference between the original information requirement (i.e., based on just the proportion of classes) and the new requirement (i.e., obtained after partitioning on A). The attribute A with the highest information gain, $\text{Gain}(A)$, is chosen as the splitting attribute at node N .

3.3.2 Decision tree pruning

When a decision tree is built, some of the branches will reflect anomalies in the training data due to noise or outliers. Pruning

addresses this problem of overfitting the data. Pruned trees tend to be smaller and less complex. C4.5 uses pruning method in which subtrees are removed from a “fully grown” tree. A subtree at a given node is pruned by removing its branches and replacing it with a leaf. The leaf is labelled with the most frequent class among the subtree being replaced. C4.5 uses pessimistic pruning, which makes use of error rate (percentage of tuples misclassified by the tree) estimates to make decisions regarding subtree pruning. The training set is used to estimate error rates. The pruning starts from the bottom of the tree. For each internal node, N , it computes the cost complexity of the subtree at N , and the cost complexity of the subtree at N if it were to be pruned (i.e., replaced by a leaf node). The two values are compared. If pruning the subtree at node N would result in a smaller cost complexity, then the subtree is pruned. Otherwise, it is kept.

In the present study, to find the optimum parameters, an exhaustive search is carried out. Two parameters of the decision tree that need to be optimized for best accuracy are: the confidence level and minimum number of examples in the leaves. Table 2 presents the results of decision tree pruning for different confidence levels and minimal number of examples (for dataset with market data upto March 7, 2010). For other confidence levels, not displayed in the table, the results were similar, with the best accuracy is achieved when the minimum number of examples in leaves is 2 at confidence level of 25%, (from table 2). The features that form the most accurate decision tree are the selected features and are used for predicting the future trend. Other features which are not selected are discarded and not considered further.

Table 2. Decision tree pruning results

Min. Ex.	Confidence Level (%)						
	5	10	25	50	75	85	95
2	67.8	67.9	68.3	68.1	68.1	68.1	68.1
10	66.3	66.2	66.2	66.0	66.0	66.0	66.0
50	63.3	63.4	63.4	63.4	63.4	63.4	63.4
100	61.9	61.5	61.5	61.5	61.5	61.5	61.5
200	55.4	55.4	55.4	55.4	55.4	55.4	55.4
300	59.4	59.4	59.4	59.4	59.4	59.4	59.4
400	46.8	46.8	47.2	47.2	47.2	47.2	47.2
600	50.0	50.0	50.0	50.0	50.0	50.0	50.0
800	45.2	45.2	45.2	45.2	45.2	45.2	45.2
900	45.2	45.2	45.2	45.2	45.2	45.2	45.2
1000	45.2	45.2	45.2	45.2	45.2	45.2	45.2

3.4 Rough Set based Trend Prediction

Rough sets are extremely useful in dealing with incomplete or imperfect knowledge. This section presents a brief overview of the rough set concepts [17] used in the present study and also presents their application for stock market trend prediction.

A rough set uses the concepts of lower and upper approximations to classify the domain of interest into disjoint categories. The lower approximation is a description of the domain objects that are known with certainty to belong to the subset of interest, whereas the upper approximation is a description of the objects that possibly belong to the subset.

3.4.1 Information and decision system

In rough set theory knowledge is represented in information tables. Each row in the table represents an object. Each column in the table represents an attribute (also referred to as feature, in this study), in the present case, each attribute is a technical index obtained from the historical SENSEX data.

An information system, Λ is defined as:

$$\Lambda = (U, A) \quad (6)$$

U = Non empty finite set of objects called the universe

A = Non empty finite set of features such that

$$\alpha: U \rightarrow V_x \quad \forall \alpha \in A \quad (7)$$

And V_x is the value set for α

Inclusion of decision features (attributes) along with the features (attributes) of the information system results in a decision system. If Q is the set of decision attributes, a decision system S can be represented as

$$S = (U, A \cup Q) \quad (8)$$

The elements of A are called conditional attributes or conditions and U is the universal set as defined in equation (6). The decision attribute for the present study is the trend of SENSEX (up trend, down trend and no trend).

3.4.2 Indiscernibility

The concept of indiscernibility is used to define equivalence classes for the objects. Given a subset of attributes $P \subseteq A$, each such subset defines an equivalence relation $IND(P)$ called an indiscernibility relation. This indiscernibility relation is defined as

$$IND(P) = \{ (x, y) \in U^2 \mid \alpha \in P, \alpha(x) = \alpha(y) \} \quad (9)$$

Equation (9) means that the subset of features, P , will define the partition of the universe into sets such that each object in a set cannot be distinguished from other objects in the set using only the features in P . The sets into which the objects are divided, are called the equivalence classes. The equivalence classes of the indiscernibility relation with respect to P are denoted $[x]_P$, where $x \in U$. The partition of U , determined by $IND(P)$, is denoted $U/IND(P)$ or U/P , which is simply the set of equivalence classes generated by $IND(P)$.

3.4.3 Set approximation

Given an information system, $\Lambda = (U, A)$, the set $X \subseteq U$, can be approximated using only the information in the set P using the P-upper and P-lower approximations of X . P-Lower approximation of X :

$$\underline{P}X = \{ x \mid x \in \bar{P} \subseteq X \} \quad (10)$$

The lower approximation is the set containing all objects for which the equivalence class corresponding to the object is a subset of the set. This set contains all objects which certainly belong to the set X . P-Upper approximation of X :

$$\bar{P}X = \{ x \mid x \in \bar{P} \cap X \neq \emptyset \} \quad (11)$$

This set contains all objects which possibly belong to the set X . Table 3 presents the number of objects in the upper and lower approximations of each class.

Table 3. Approximation results

Class	No. of objects	Lower Approx.	Upper Approx.
Up	734	725	742
Down	194	192	196
Notrend	697	687	708

3.4.4 Reducts

A reduct is a minimal representation of the original dataset. The concepts of positive region, feature dependency and significance are used for computation of reducts. If P and Q are sets of features inducing equivalence relations over U , then the positive region comprises all objects of U that can be classified to classes of U/Q using the information contained within features P . The positive region is denoted by $POS_P(Q)$. For $P, Q \subseteq A$, the dependence of Q on P is given by

$$k = \gamma_P(Q) = |POS_P(Q)| / |U| \quad (12)$$

The dependency k can take values from 0 (meaning Q is independent of P) to 1 (meaning that Q is completely dependent on P). The significance of feature can be determined by calculating the change in dependency when a feature is removed from the set of considered features. The higher the change in dependency, the more significant is the attribute. Considering the sets P and Q and an attribute $x \in P$, the significance of feature x upon Q is defined by

$$\sigma_P(Q, x) = \gamma_P(Q) - \gamma_{P-\{x\}}(Q) \quad (13)$$

A reduct is a minimal subset of the initial feature set such that no features can be removed from the subset without affecting the dependency degree. A given dataset may have many reduct sets, and the collection of all reducts is denoted by

$$R = \{ X \subseteq P; \gamma_X(D) = \gamma_P(D); \gamma_{X-\{a\}}(D) \neq \gamma_X(D), \forall a \in X \} \quad (14)$$

The core is a subset of the reduct set R . The elements of core are those features that cannot be eliminated without introducing more contradictions to the dataset.

The following five reducts were obtained:

1. Volume, MACD Line, Nine-period moving average, Fast stochastic %K, Fast stochastic %D, Acceleration, Momentum, William's %R, Negative volume index, RSI, Bollinger-mid, Highest High, On balance volume, Price rate of change, Price-volume trend, Typical Price, William's accumulation/distribution.

2. Volume, MACD Line, Nine-period moving average, Fast stochastic %K, Fast stochastic %D, Acceleration, Momentum, William's %R, Negative volume index, RSI, Bollinger-mid, Highest High, Lowest Low, On balance volume, Price rate of change, Price-volume trend, William's accumulation/distribution.

3. Volume, MACD Line, Nine-period moving average, Fast stochastic %K, Fast stochastic %D, Acceleration, Momentum, William's %R, Negative volume index, Positive volume index, RSI, Bollinger-mid, Highest high, On balance volume, Price

rate of change, Price-volume trend, William’s accumulation/distribution.

4. Open, Volume, MACD Line, Nine-period moving average, Fast stochastic %K, Fast stochastic %D, Acceleration, Momentum, William’s %R, Negative volume index, RSI, Bollinger mid, Highest high, on balance volume, Price rate of change, Price-volume trend, William’s accumulation/distribution.

5. Volume, MACD Line, Nine-period moving average, Fast stochastic %K, Fast stochastic %D, Acceleration, Momentum, William’s %R, Negative volume index, Positive volume index, RSI, Bollinger-mid, Highest high, Lowest Low, Price rate of change, Price-volume trend, William’s accumulation/distribution.

There are fifteen attributes in the core, namely, Volume, MACD Line, nine period moving average, fast stochastic %K, fast stochastic %D, acceleration, momentum, william’s %R, negative volume index, RSI, bollinger-mid, highest high, price rate of change, price-volume trend, William’s A/D.

3.4.5 Discernibility matrices

A discernibility matrix of Λ is a symmetric $n \times n$ matrix with entries

$$C_{ij} = \alpha \in A \mid \alpha(x_i) \neq \alpha(x_j) \text{ for } i, j = 1, 2, \dots, n. \quad (15)$$

The entries for each object are thus the attributes that are needed in order to discern object i from object j .

3.4.5 Decision rules

An approximation of decision Q , for decision system

$$S = (U, A \cup Q) \quad (16)$$

can be done by constructing the decision relative discernibility matrix of S . The decision –relative discernibility matrix of S gives information about how to discern an object from objects belonging to another decision class. If $G(A) = c_{ij}$ is the discernibility matrix of a decision system S , the decision relative discernibility matrix of S is defined as

$$G^r(A) = (c_{ij}^r) \quad (17)$$

Assuming $c_{ij}^r = 0$ if $q(x_i) = q(x_j)$ and $c_{ij}^r = C_{ij} - q$ otherwise.

From the reducts computed from the discernibility matrix, the decision rules are generated for classification of the objects. In the present study, a total of 199 rules were generated. The classification accuracy was evaluated using a 10-folds cross-validation method and was found to be 90.22%. The confusion matrix is given in table 4.

Table 4. Hybrid decision tree-roughset confusion matrix

	No trend	Up trend	Down trend
No trend	614	53	30
Uptrend	49	684	1
Down trend	26	0	168

A stand-alone rough set based trend prediction system without feature selection was also designed and evaluated. A total of 202 decision rules were obtained and the classification accuracy obtained was 88.18 %.

4. ARTIFICIAL NEURAL NETWORK AND NAIVE BAYES BASED PREDICTORS

4.1 Artificial Neural Network based Trend Predictor

Artificial neural networks (ANNs) have been used widely for predicting stock markets. In the present study, a simple multilayer feedforward neural network trained using backpropagation learning is used for predicting the next days’ trend of the SENSEX. All the features considered for the study initially, are used as inputs with each feature forming one input neuron. The inputs are normalized to -1 to +1 range for better performance. Training parameters which gave the best performance are :

Learning rate $\varepsilon = 0.3$

Momentum $\alpha = 0.2$

Number of Hidden Layers = 1

Number of neurons in the input layer = 30

Number of neurons in the hidden layer = 30

Number of neurons in the output layer = 3 (one each for Up, Down and No trend)

A classification accuracy of 77.66% was obtained.

The prediction results are presented in the form of a confusion matrix in table 5.

Table 5: ANN trend predictor confusion matrix

	No trend	Uptrend	Downtrend
No trend	597	100	0
Uptrend	69	665	0
Downtrend	194	0	0

4.2 Naive Bayes based Trend Predictor

A naive Bayes based trend prediction system is also designed and evaluated. A naive Bayes based classifier/prediction system is much easier to design when compared to neural network based or rough set based classifier. Hence, the naive Bayes based prediction system is designed and evaluated to showcase the improvement in accuracy that is brought about by the use of neural network based and the rough set based prediction systems. Bayesian classifiers are statistical classifiers. They can predict class membership probabilities, such as the probability that a given tuple belongs to a particular class. Naïve Bayesian classifiers assume class conditional independence, that is, the effect of an attribute value on a given class is considered independent of the values of the other attributes. It is made to simplify the computations involved and, in this sense, is considered “naïve.” Let \mathbf{X} be a data tuple (referred to as ‘evidence’), described by a set of n attributes such that $\mathbf{X} = (x_1, x_2, x_3, x_4, \dots, x_n)$. Assuming that there are m classes, C_1, C_2, \dots, C_m , the classifier will predict that \mathbf{X} belongs to the class having the highest posterior probability, conditioned on \mathbf{X} . By Bayes’ theorem

$$P(C_i | \mathbf{X}) = \frac{P(\mathbf{X} | C_i) P(C_i)}{P(\mathbf{X})} \quad (18)$$

As $P(\mathbf{X})$ is constant for all classes, only $P(\mathbf{X} | C_i)P(C_i)$ need be maximized. The naive assumption of class conditional independence implies

$$P(\mathbf{X} | C_i) = \prod_{k=1}^n P(x_k | C_i) \quad (19)$$

In the present study, the attributes are continuous and hence, are assumed to have a Gaussian distribution with a mean μ and standard deviation σ ,

$$P(x_k | C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) \quad (20)$$

$$\text{Where } g(x, \mu, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

In order to predict the class label of \mathbf{X} , $P(\mathbf{X}|C_i)P(C_i)$ is evaluated for each class C_i . The naïve Bayesian classifier predicts that tuple \mathbf{X} belongs to the class C_i if and only if $P(C_i | \mathbf{X}) > P(C_j | \mathbf{X})$ for $1 \leq j \leq m, j \neq i$.

In the present study, three classes are considered: $C_1 = \text{Up}$, $C_2 = \text{down}$ and $C_3 = \text{no trend}$. Each \mathbf{X} is the set of all the attributes for one day.

On application of the Naive Bayes classification technique, the accuracy obtained was 72.36% and the confusion matrix is given in table 6.

Table 6: Naive Bayes trend predictor confusion matrix

	No trend	Up trend	Down trend
No trend	414	196	87
Uptrend	111	623	0
Down trend	53	2	139

5. CONCLUSIONS

The design of rough set based stock market trend prediction system for predicting the one-day-ahead trend of the SENSEX is presented in this paper. Features are extracted from the historical SENSEX data. Extracted features are used to generate the prediction rules using rough set theory. It is observed that the proposed hybrid decision tree-rough set based trend prediction system produces an accuracy of 90.22%. The stand-alone rough set based trend prediction system, without any feature selection, produced an accuracy of 88.18 %. The artificial neural network based prediction system, without feature selection, yielded an accuracy of 77.66%. Both these systems are found to offer better performance when compared to a simple Naïve bayes based trend prediction system which provided a trend prediction accuracy of 72.36%. It is seen that automated feature selection employed in the present study significantly improves the trend prediction accuracy. It can be concluded from the present study that soft computing based techniques, particularly the hybrid decision tree-rough set based trend prediction system is well suited for stock market trend prediction.

6. REFERENCES

- [1] Fama, E. F.,(1970), 'Efficient capital markets: A review of theory and empirical work', Journal of Finance, 25, pp. 383–417.
- [2] Atsalakis G. S., and Valavanis K. P., (2009), 'Surveying stock market forecasting techniques – part II: soft computing methods', Expert Systems with Applications, vol.36, pp. 5932–5941.
- [3] Saad E. W., Prokhorov D.V., and Wunsch, D.C., (1998), 'Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks', IEEE Transactions on Neural Networks, Vol. 9, No. 6, pp. 1456-1470.
- [4] Lee, C-T., and Chen,Y-P. 2007. The efficacy of neural networks and simple technical indicators in predicting stock markets. In Proceedings of the International Conference on Convergence Information Technology, pp.2292-2297.
- [5] Kuo, M-H., and Chen, C-L.2006. An ETF trading decision support system by using neural network and technical indicators. In Proceedings of the International Joint Conference on Neural Networks, pp. 2394-2401.
- [6] Nagarajan,V., Wu,Y., Liu,M. and Wang Q-G. 2005. Forecast Studies for Financial Markets using Technical Analysis. In Proceedings of the International Conference on Control and Automation (ICCA2005), pp. 259-264.
- [7] Bansal, Archit, Mishra ,Kaushik, Pachouri, Anshul, (2010), 'Algorithmic Trading (AT) - Framework for Futuristic Intelligent Human Interaction with Small Investors', International Journal of Computer Applications, vol. 1, no. 21, pp.01-05.
- [8] Chang, P.-C. , Liu, C.-H. , Lin, J.-L. , Fan, C.-Y. , & Celeste, S.P. Ng., (2009) , 'A neural network with a case based dynamic window for stock trading prediction', Expert Systems with Applications, vol.36, pp.6889–6898.
- [9] Setty, Venugopal D., Rangaswamy, T.M. and Subramanya, K.N.,(2010), 'A Review on Data Mining Applications to the Performance of Stock Marketing', International Journal of Computer Applications, vol. 1, no. 3,pp.33-43.
- [10] Wang, J-L. , and Chan, S-H.,(2006), 'Stock market trading rule discovery using two-layer bias decision tree', Expert Systems with Applications, 30, pp.605–611.
- [11] Wu, M-C., Lin, S-Y., & Lin, C-H.,(2006) , 'An effective application of decision tree to stock trading', Expert Systems with Applications, 31, pp.270–274.
- [12] Huang, K.Y, and Jane, C.-J., (2009), 'A hybrid model for stock market forecasting and portfolio selection based on ARX, grey system and RS theories', Expert Systems with Applications, 36, pp.5387–5392.
- [13] Teoh, H. J., Cheng,C-H., Chu, H-H., and Chen, J-S., (2008), 'Fuzzy time series model based on probabilistic approach and rough set rule induction for empirical research in stock markets', Data & Knowledge Engineering, 67, pp.103–117.
- [14] Eng,W.F. 1988 The Technical Analysis of Stocks, Options & Futures- Advanced Trading Systems and Techniques. Vision Books,India.
- [15] www.trendwatch.co.uk
- [16] Han, Jiawei and Kamber, Micheline 2006 Data Mining:Concepts and Techniques-Second Edition, Morgan Kaufmann,USA, pp 291-315.
- [17] Jensen,R., & Shen, Q. 2008 Computational Intelligence And Feature Selection-Rough and Fuzzy Approaches, John Wiley & Sons.