A Framework to Generate Features with Minimum Variance: Evaluation in Indian Stock Market

Puneet Misra[1], Siddharth Chaurasia[1*]
[1]*Department of Computer Science, University of Lucknow*
(E-mail: puneetmisra@gmail.com; siddharth515@gmail.com[*])

Abstract**—** Prediction of the stock market has been one of the most attempted forecasting problems. Often, technical analysis is used to generate features for short-term prediction of the market. There are hundreds of features which can be used to gauge the market movement. Selection of features to use is a challenging problem. This paper proposes a framework for the selection of the pre-defined number of features by usage of automatic feature selection techniques. Proposed framework makes use of four different Recursive Feature Elimination, Mutual Information Gain, Random Forest and Extra Tree techniques and selects the final set by majority voting.

Keywords**—***Feature Selection; Machine Learning; Random Forest; Mutual Information, Recursive Feature Elimination;*

I. INTRODUCTION

Stock market price prediction is regarded as a challenging task of the financial time series prediction process since the financial market is a complex, evolutionary, and non-linear dynamic system [1]. Recently, the field of market forecasting has envisaged usage of AI-based techniques like Machine learning. One of the most important factors governing the machine learning predictions is the features used. Learning is easy if a set of independent features is selected that correlate well with the class. Conversely, if the class is a very complex function of the features, it may not be able to learn it [2]. Thus, the problem of feature selection is tantamount for any good execution. Feature selection can provide many probable benefits like facilitation of data understanding, reducing the measurement and storage requirements, minimizing training and utilization times, defying the curse of dimensionality to improve prediction performances [3].

Historical market data comes in the form of Open, High, Low, Close and Volume (OHLCV). Though few studies have used only the raw data for learning as in [4], [5], often the raw data is not in a form that is amenable to learning but can be used to construct features that are more informative and can be used to uncover the underlying relationships. The quantitative technical analysis in financial trading uses mathematical and statistical tools to provide the insight into the market movement [6]. There are hundreds of representations that are available in the form of technical indicators. Each indicator may produce different signals like buy, hold or sell depending on their definitions.

Feature selection is a pre-processing step of any ML or data mining implementation. It is used to filter redundant and irrelevant features and select an optimal set. Feature selection results in a simpler model, easier interpretation, and faster induction and structural knowledge [7]. Although many studies have claimed feature selection to be the key process in stock prediction modelling, identifying more representative features and improving stock prediction are challenging issues that need to be considered. There are many feature selection algorithms like PCA which transform the original features to more effective and smaller set of features while others are filtered techniques which aim at selecting effective feature set. This paper uses latter techniques as it not only reduces the feature set but helps in understanding the most relevant features too.

Similar work was done in [8] where multiple techniques were combined for feature selection. The aim was to filter out unrepresentative variables from a given dataset for effective prediction. Since different feature selection methods result in the selection of different features authors combined the multiple feature selection techniques to identify more representative variables for better prediction. The study used Principal Component Analysis (PCA), Genetic Algorithm (GA) and Decision Trees (CART). Some other works that have considered feature selection step on the initial set of variables are [9],[10], [11].

The idea of combining multiple feature selection methods is derived from classifier ensembles (or multiple classifiers) [12]. The ensemble approach aims to obtain accurate classifier by combining weak or less accurate ones. They are intended to improve the classification performance of a single classifier. Similarly, in the case of feature selection, the aim of combining more than one filter approach is to individual approach complemented by other approaches. The final set of features should be able to provide a more representative set of features. Thus, the assumption is that the combination can complement the errors made by the individual selections on different parts of the input space.

This paper proposes a framework for an efficient selection of features which minimizes the variations in the accuracy as the number of features varies thus signifying the stability of the selection. Paper uses four individual feature selection methods: recursive feature elimination (RFE), Mutual Information (MI), random forest (RF) and Extra Tree (ET). Paper empirically shows that the feature set generated by the proposed framework gives minimum variation in f-score and accuracy when more than five features are selected.

II. FEATURE SELECTION TECHNIQUES

This section first describes different feature selection techniques used in the paper that forms the basis of the proposed framework MVFS. In the second part, the MVFS framework is discussed and described.

A. *Recursive Feature Elimination (RFE)*

Recursive feature elimination (RFE) uses an external estimator that assign weights to features, e.g. Linear model, logistic regression. The goal of is to select features recursively by considering smaller and smaller sets of most appropriate ones. First, the estimator is trained on the complete set of features, and the importance of each feature is obtained. Then, the least significant features are pruned from the current set of features. The procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached [13].

Thus, RFE is implemented through a backwards selection of predictors based on predictor importance ranking. The predictors are ranked, and the less important ones are sequentially eliminated. The goal of the last step is to find a subset of predictor that can result in accurate predictions without overfitting. In the experiments (Logistic Regression) LoR-RFE is used for each ranking.

B. *Mutual Information (MI)*

A good feature representation should have independent and non-correlated feature set. Mutual information is used to identify related features with the aim to have features set comprised of disjoint features. Feature selection by measuring mutual information uses the degree of relatedness between data sets to rank usefulness of the features. MI detects any relationship between data sets like mean values or the variances or higher moments.

Mutual information (MI) between two random variables is a value that measures the dependency between the variables. It is equal to zero if the two random variables are independent, and higher values mean higher dependency. The implementation in the experiments is for selection of n best features. It is based on the entropy estimation from k-nearest neighbors distances as described in [14] while the idea was proposed in [15].

C. *Random Forest (RF)*

Random Forest is a popular approach for classification and regression problems. They can also be used as a means to distinguish relevant from irrelevant variables in variable selection approaches [16].

The ranking by relevance by RF is possible because it is an ensemble of decision trees and tree-based strategies naturally rank by improvement in the purity of the node. Every node in the decision trees signifies a condition on a single feature, such that each split of the dataset into two, makes the similar response values end up in the same set. The measure based on which the optimal condition is chosen is called impurity. Thus, while training a tree, at each step the increase and decrease in impurity can be computed. For a forest, the impurity decrease from each feature can be averaged, and the features are ranked according to this measure.

The impurity-based reduction is biased towards preferring variables with more categories. Moreover, if the dataset has two or more correlated features, then for the split hardly any difference is considered between them.

D. *Extremely Randomized Trees (ET)*

Similar to RF, extremely randomized trees can also be used to rank the features. The Extra-Trees algorithm builds an ensemble of the unpruned decision or regression trees according to the top-down procedure. Its two main differences with other tree-based ensemble methods are that it splits nodes by choosing cut-points entirely at random and that it uses the whole learning sample (rather than a bootstrap replica) to grow the trees [17]. Thus, ET provides a different way of ranking the features.

E. *Proposed: Majority Voting for Feature Selection (MVFS)*

Proposed framework (MVFS) applies ranking by majority voting on the ranked feature set generated by the four individual feature selection methods. The final rank of the features is the average of the individual ranks. The rationale behind the averaging is first to reduce dependency on one single method which often carries specific challenges, e.g. if feature selection is made by RF, strong features can end up with low scores if there are correlated features and the method can be biased towards variables with many categories. Combining multiple selection techniques provides a way of covering one’s weakness by the strength of other.

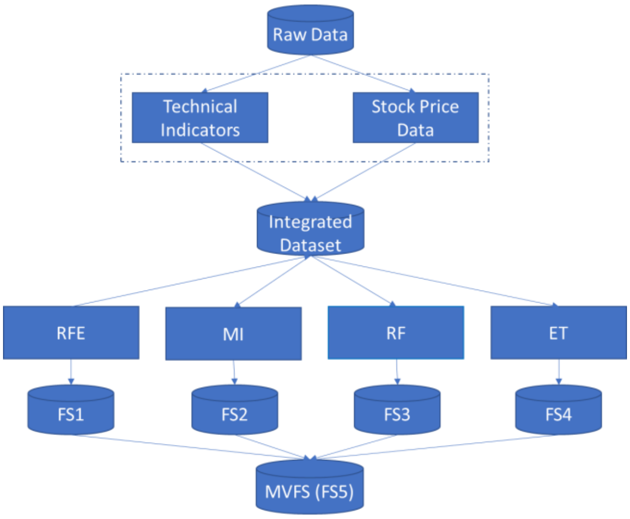

Fig. 1: Proposed Framework for Final Feature Selection

We name the generated features sets as FS1 to FS5 using each of the methods. FS5 is generated using the proposed approach MVFS.

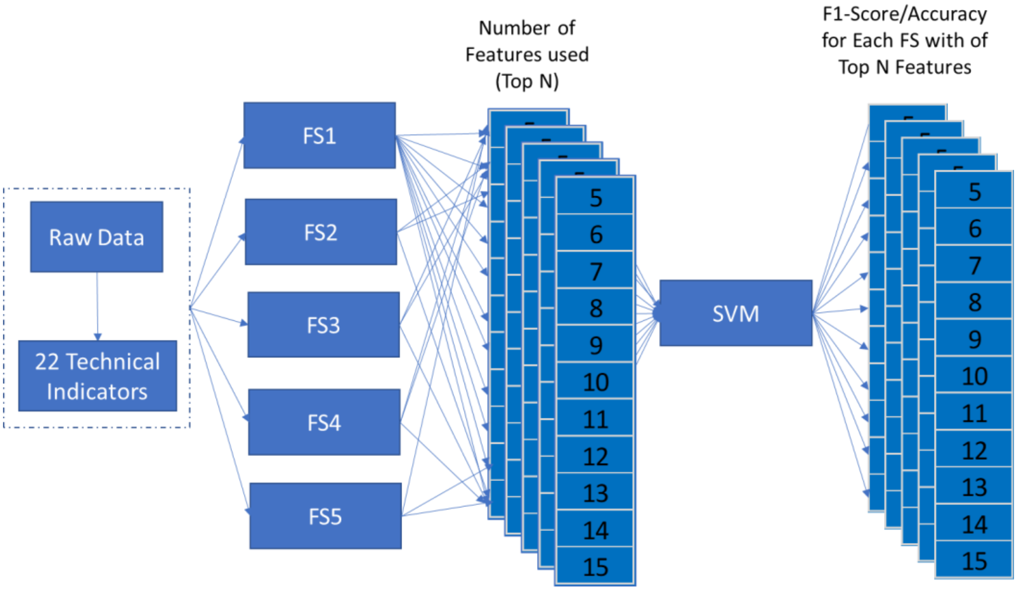

Fig. 2: Experimental Setup

III. EXPERIMENTAL SETUP

Experimentation in this study has been carried out for identifying the best feature selection technique. A set of 22 technical indicators are initially generated using definitions from [18] and [19]. Though there exist studies in the literature that have used feature sets more than 50 like in [20], [21]; Still Atsalakis noticed that most of the studies incorporate features in the range of 10 [22]. Based on this, feature range of 5 to 15 is tested to identify the most stable feature selection approach. Stability has been measured by the variance in the f-score. More the variance, less the stability of the selected features.

Experiments are conducted on two most popular stock indices from Indian stock market: Nifty 50 and S&P BSE Sensex. Data used is from 2004 to 2017, i.e. of 14 years and has been retrieved from www.nseindia.com and www.bseindia.com. SVM with RBF kernel is used to make the predictions. Reason for the choice of SVM is driven by observations in literature where it has been acclaimed as one of the best methods for predictions [23].

Table 1 details the initial feature generation process using technical analysis. The continuous values generated using technical analysis are converted to discrete trend signals using the definitions in table 2. Moreover, the same indicator is used to generate multiple signals based on the period used as in [19]. Thus three indicators generate 12 signals, 2 for momentum, five signals for moving averages and OBV.

Different feature sets (FS) are created for all the feature selection techniques by ranking them from top to bottom. An attempt is made to predict the next day direction of the high price for stock indices with each of FS. Choice of the high price as predictor variable is governed by an observation made in [24] which points out that high prices do not carry random noise which is part of closing prices due to day-end effect. Inputs to the prediction algorithm comprise of different feature counts starting from 5 till 15. Thus, in total 10x5, i.e. 50 executions are made for the prediction algorithm.

Table 1: Technical Indicators Used in Experiments as in [18] and [19]

Name of Indicators	Formulas
Simple n(10) day Moving Average	$\frac{C_t+C_{t-1}+...+C_{t-9}}{n}$
Weighted n(10) day Moving Average	$\frac{(10)C_t+(9)C_{t-1}+...+C_{t-9}}{n+(n-1)+...+1}$
Momentum	C_t-C_{t-9}
Stochastic K%	$\frac{C_t-LL_{t-(n-1)}}{HH_{t-(n-1)}-LL_{t-(n-1)}}*100$
Stochastic D%	$\frac{\sum_{i=0}^{n-1} K_{t-i}}{10}\%$
Relative Strength Index (RSI)	$100-\frac{100}{1+(\sum_{i=0}^{n-1} UP_{t-i}/n)/(\sum_{i=0}^{n-1} DW_{t-i}/n)}$
Moving Average Convergence Divergence (MACD)	$MACD(n)_{t-1}+\frac{2}{n+1}*(DIFF_t-MACD(n)_{t-1})$
Larry William R% (WR)	$\frac{H_n-C_t}{H_n-L_n}*100$
Accumulation/Distribution (A/D) Oscillator	$\frac{H_t-C_{t-1}}{H_t-L_t}$
Commodity Channel Index (CCI)	$\frac{M_t-SM_t}{0.015D_t}$
On Balance Volume (OBV)	$OBV=\sum_{j=1}^{t} Vol_{i,j}\times D_{i,j}$

Ct-Closing price, Ht-High Price, Lt-Low Price at time t. $Diff_t=EMA(12)_t-EMA(26)_t$ *EMA is Exponential Moving Average,* $EMA(k)_t=EMA(k)_{t-1}+\alpha\times(C_t-EMA(k)_{t-1}), \alpha=2/(k+1)$ *k=time period of k day EMA,* LL_t *and* HH_t *imply lowest low and highest high in last t days.* $M_t=(H_t+L_t+C_t)/3$, $D_t=(\sum_{i=1}^{n}|M_{t-i+1}-SM_t|)/n$ UP_t *means upward price change and* DW_t *is the downward price change at time t .* $OBV_{i,j}$ *is trading volume during period j and Di, j is binary variable signifying the change from previous day*

Table 2: Generation of Signals for discretization before applying feature selection

Name of indicators	**Rules for Signals**
Signal Generation Based on [18]	
Simple n (10) day Moving Average (SMA)	$If\ (SMA_t \geq CP_t) \Rightarrow +1\ else\ -1$
Weighted n(10)-day Moving Average (WMA)	$If\ (WMA_t \geq CP_t)\ \Rightarrow +1\ else\ -1$
Momentum (Mom)	$If\ (Mom_t \geq 0) \Rightarrow +1\ else\ -1$
Stochastic K% (STCK)	$If\ (STCK_t \geq\ STCK_{t-1}) \Rightarrow +1\ else\ -1$
Stochastic D% (STCD)	$If\ (STCD_t \geq\ STCD_{t-1}) \Rightarrow +1\ else\ -1$
Moving Average Convergence Divergence (MACD)	$If\ (MACD_t \geq MACD_{t-1}) \Rightarrow +1\ else\ -1$
Larry William's R% (WR)	$If\ (WR_t \geq WR_{t-1}) \Rightarrow +1\ else\ -1$
A/D (Accumulation/Distribution) Oscillator (AD)	$If\ (AD_t \geq AD_{t-1}) \Rightarrow +1\ else\ -1$
Relative Strength Index (RSI)	$If\ (RSI > 70\ ) \Rightarrow -1; If\ \ (RSI < 30) \Rightarrow +1;$ $If\ \ (30 \leq RSI \leq 70)\ and\ \ (RSI_t >\ RSI_{t-1}) \Rightarrow +1\ else\ -1$
Commodity Channel Index (CCI)	$If\ (CCI > 200\ ) \Rightarrow -1;\ If\ (CCI\ \ < -200) \Rightarrow +1;$ $If\ (-200 <= CCI <= 200)\ and\ \ (CCI_t >\ CCI_{t-1}) \Rightarrow +1\ else\ -1$
Signal Generation Based on [19]	
Momentum (for m = 9 and 12)	$S_{i,t} =\ 1\ if\ \ P_{i,t} \geq P_{i,t-m};\ 0\ if\ \ P_{i,t} < P_{i,t-m}$
Moving Average ($MA_{i,t}^{m} = \frac{1}{m}\sum_{h=0}^{m-1} P_{i,t-h}$) ($m = s,\text{l}\ \ s = short\ \ MA, l = Long\ \ MA$)	$S_{i,t} = \begin{cases} 1\ if\ \ MA_{i,t}^{s} \geq MA_{i,t}^{l} \\ 0\ if\ \ MA_{i,t}^{s} < MA_{i,t}^{l} \end{cases}, s = 1,2,3; l = 9,12$
On Balance Volume $OBV = \sum_{j=1}^{t} Vol_{i,j} \times D_{i,j}$ $m = s,\text{l}\ \ s = short\ \ MA, l = Long\ \ MA$	$S_{i,t} = \begin{cases} 1\ if\ \ MA_{i,t}^{OBV,s} \geq MA_{i,t}^{OBV,l} \\ 0\ if\ \ MA_{i,t}^{OBV,s} \geq MA_{i,t}^{OBV,l} \end{cases}, s = 1,2,3; l = 9,12$

IV. EVALUATION METRICS AND RESULTS

Accuracy and F-score are used to evaluate the performance of the prediction models. Computation of both the evaluation measures requires estimating Precision and Recall which are evaluated from True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). Formulation of the metrics used is detailed in table 3.

Table 2: Evaluation Measures Used

Metric	**Formula**
Precision (P)	TP/(TP+FP) & TN/(TN+FN)
Recall (R)	TP/(TP+FN) & TN/(TN+FP)
F-Score	(2*P*R)/(P + R)
Accuracy	(TP+TN)/(TP+TN+FP+FN)

TN-True Negative, TP-True Positive, FP-False Positive, FN-False Negative

Table 4 and 5 display the f-score and accuracies when a different number of features are used for each of the feature selection methods used in this paper along with the proposed approach with the Nifty dataset.

Table 4: Prediction F-Score for Different number of Feature for Nifty

Number of Features	**ET**	**RF**	**RFE**	**MI**	**MVFS**
5	0.7034	0.6976	0.6993	0.7027	0.7034
6	0.7012	0.6993	0.6990	0.7021	0.7030
7	0.7031	0.7022	0.6992	0.7033	0.7056
8	0.7021	0.7031	0.6991	0.7041	0.7045
9	0.7023	0.7045	0.7008	0.7041	0.7051
10	0.7020	0.7048	0.7030	0.7052	0.7050
11	0.7028	0.7052	0.7022	0.7057	0.7040
12	0.7040	0.7041	0.7053	0.7051	0.7048
13	0.7037	0.7041	0.7053	0.7051	0.7041
14	0.7040	0.7044	0.7044	0.7048	0.7041
15	0.7037	0.7041	0.7053	0.7045	0.7038
Variance	**0.0028**	**0.0076**	**0.0063**	**0.0036**	**0.0026**

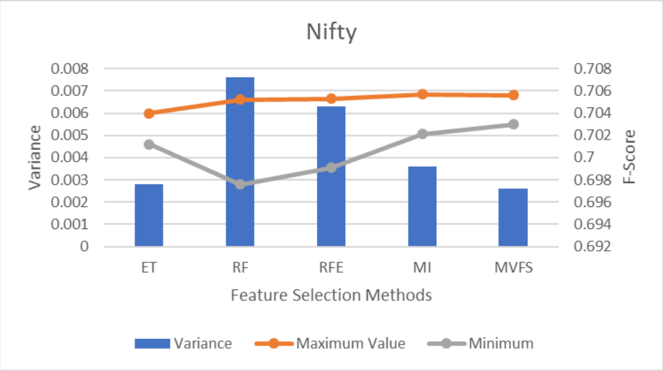

Fig. 3: Variance, Minimum & Maximum F-Scores for Each Feature Selection Method for Nifty 50

Table 5: Accuracy for Different Feature Counts for Nifty

Number of Features	ET	RF	RFE	MI	MVFS
5	0.7038	0.6977	0.6991	0.7047	0.7035
6	0.7027	0.6996	0.6988	0.7041	0.7038
7	0.7041	0.7027	0.6991	0.7052	0.7035
8	0.7033	0.7041	0.6991	0.7060	0.7074
9	0.7035	0.7063	0.7010	0.7060	0.7063
10	0.7030	0.7066	0.7041	0.7072	0.7069
11	0.7041	0.7069	0.7035	0.7077	0.7069
12	0.7058	0.7058	0.7069	0.7072	0.7058
13	0.7055	0.7058	0.7069	0.7072	0.7066
14	0.7063	0.7060	0.7060	0.7069	0.7058
15	0.7058	0.7058	0.7072	0.7066	0.7058
Variance	**0.0036**	**0.0092**	**0.0084**	**0.0036**	**0.0039**

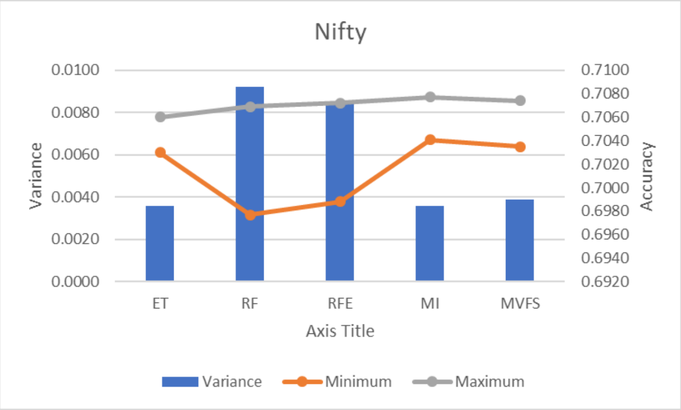

Fig. 4: Variance, Minimum & Maximum Accuracies for Each Feature Selection Method for Nifty 50

Similarly, Table 6 and 7 present the f-scores and accuracies for S&P BSE Sensex.

Table 6: F-Scores for Different Feature Counts for S&P Sensex

Number of Features	ET	RF	RFE	MI	MVFS
5	0.6999	0.6979	0.6954	0.6993	0.6999
6	0.6991	0.6991	0.6953	0.6994	0.6995
7	0.6989	0.7001	0.6981	0.6983	0.7006
8	0.7011	0.7014	0.6979	0.6993	0.7008
9	0.7001	0.701	0.6951	0.6988	0.6998
10	0.7002	0.7015	0.698	0.7013	0.7013
11	0.7001	0.7012	0.6993	0.6999	0.7013
12	0.7019	0.7014	0.6996	0.6995	0.7012
13	0.7013	0.7010	0.6996	0.6996	0.7010
14	0.6998	0.7018	0.7002	0.6996	0.7000
15	0.6992	0.7013	0.7005	0.7014	0.7000
Variance	**0.0030**	**0.0027**	**0.0054**	**0.0031**	**0.0018**

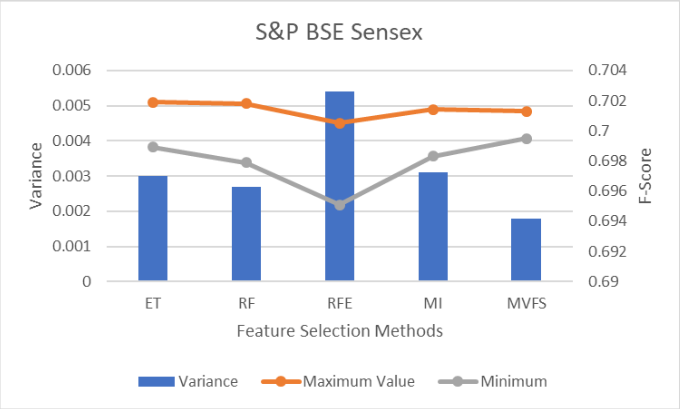

Fig. 5: Variance, Minimum & Maximum F-Scores for Each Feature Selection Method for S&P BSE Sensex

Figure 3, 4, 5 and 6 displays the variance shown by each of the techniques for nifty and Sensex datasets along with maximum and minimum prediction scores obtained for each of the feature selection techniques when number of features presented ranged from 5 to 15. MVFS provided the least variance in prediction accuracies as the number of features changed. While, at the same time MVFS was able to achieve maximum f-score values nearly equal to the best score obtained by features set generated by other methods. Thus, indicating that stability is not at the cost of prediction accuracy. Rather, MVFS was able to provide best minimum prediction score thus helping in minimizing the variance.

Table 7: Accuracies for Different Feature Counts for S&P Sensex

Number of Features	ET	RF	RFE	MI	MVFS
5	0.7001	0.6977	0.6951	0.6998	0.7001
6	0.6995	0.6992	0.6951	0.7001	0.7001
7	0.6995	0.7004	0.6980	0.6989	0.6998
8	0.7016	0.7024	0.6980	0.6998	0.7010
9	0.7007	0.7016	0.6951	0.6992	0.7013
10	0.7007	0.7030	0.6983	0.7016	0.7004
11	0.7007	0.7027	0.6998	0.7004	0.7010
12	0.7024	0.7018	0.7001	0.7001	0.7018
13	0.7018	0.7016	0.7001	0.7001	0.7016
14	0.7004	0.7024	0.7007	0.7001	0.7007
15	0.6998	0.7018	0.7010	0.7018	0.7007
Variance	**0.0029**	**0.0053**	**0.0059**	**0.0029**	**0.0020**

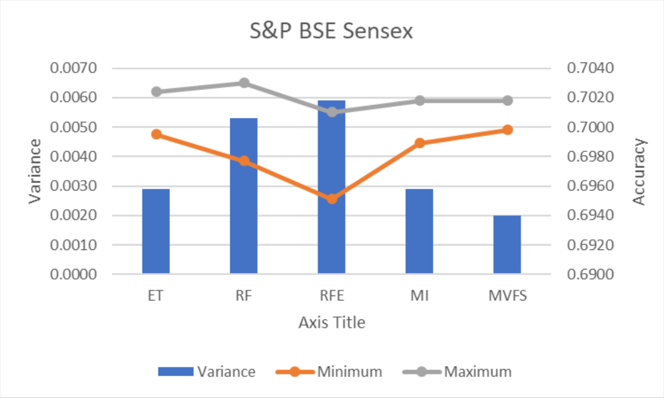

Fig. 4: Variance, Minimum & Maximum Accuracies for Each Feature Selection Method for S&P BSE Sensex

V. CONCLUSION

Feature selection technique aims to find minimal feature set which has the highest correlation with the prediction value and at the same time has the least correlation among the various attributes. Most automated feature selection techniques make use of some concept to the first rank and then filter out the features till desired numbers of features are achieved.

The paper proposed a majority voting scheme on the four feature selection methods when applied to the financial market domain. The experimental results show that the proposed approach MVFS of coalescing multiple feature selection methods can provide stable feature sets which show less variance as compared to the use of individual feature selection methods. Moreover, the stability is not obtained at the cost of performance as highest predictions obtained by MVFS is in similar range as compared to other feature selection methods. Rather, MVFS is able to minimize the drawdown or dips in the performance.

Above finding corresponds to the success of classifier ensembles, which is based on the diversity of individual classifier. That is, the ways of selecting features by RFE, MI, RF and ET individually are different, which can make the selected features by these four methods much diversified.

REFERENCES

[1] Y. S. Abu-Mostafa and A. F. Atiya, “Introduction to financial forecasting,” *Appl. Intell.*, vol. 6, no. 3, pp. 205–213, 1996.

[2] P. Domingos, “A few useful things to know about machine learning,” *Commun. ACM*, vol. 55, no. 10, p. 78, 2012.

[3] I. Guyon and A. Elisseeff, “An Introduction to Variable and Feature Selection,” *J. Mach. Learn. Res.*, vol. 3, no. 3, pp. 1157–1182, 2003.

[4] S. Jeon, B. Hong, and V. Chang, “Pattern graph tracking-based stock price prediction using big data,” *Futur. Gener. Comput. Syst.*, vol. 80, pp. 171–187, 2018.

[5] X. Pang, Y. Zhou, P. Wang, W. Lin, and V. Chang, “An innovative neural network approach for stock market prediction,” *J. Supercomput.*, 2018.

[6] E. A. Gerlin, M. McGinnity, A. Belatreche, and S. Coleman, “Evaluating machine learning classification for financial trading: An empirical approach,” *Expert Syst. Appl.*, vol. 54, pp. 193–207, 2016.

[7] H. Liu and R. Setiono, “Feature selection via discretization,” *IEEE Trans. Knowl. Data Eng.*, vol. 9, no. 4, pp. 642–645, 1997.

[8] C. F. Tsai and Y. C. Hsiao, “Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches,” *Decis. Support Syst.*, vol. 50, no. 1, pp. 258–269, 2010.

[9] C. L. Huang and C. Y. Tsai, “A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting,” *Expert Syst. Appl.*, vol. 36, no. 2 PART 1, pp. 1529–1539, 2009.

[10] Y. Chen and Y. Hao, “A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction,” *Expert Syst. Appl.*, vol. 80, pp. 340–355, 2017.

[11] B. Weng, M. A. Ahmed, and F. M. Megahed, “Stock market one-day ahead movement prediction using disparate data sources,” *Expert Syst. Appl.*, vol. 79, pp. 153–163, 2017.

[12] J. Kittler, M. Hater, and R. P. W. Duin, “Combining classifiers,” *Proc. - Int. Conf. Pattern Recognit.*, vol. 2, no. 3, pp. 897–901, 1996.

[13] F. Pedregosa, R. Weiss, and M. Brucher, “Scikit-learn : Machine Learning in Python,” vol. 12, pp. 2825–2830, 2011.

[14] B. C. Ross, “Mutual information between discrete and continuous data sets,” *PLoS One*, vol. 9, no. 2, 2014.

[15] L. F. Kozachenko and N. N. Leonenko, “Sample Estimate of the Entropy of a Random Vector,” *Probl. Peredachi Inf*, vol. 23, no. 2,

pp. 9–16, 2018.

[16] A. Hapfelmeier and K. Ulm, "A new variable selection approach using Random Forests," *Comput. Stat. Data Anal.*, vol. 60, no. 1, pp. 50–69, 2013.

[17] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.

[18] Y. Kara, M. Acar Boyacioglu, and Ö. K. Baykan, "Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5311–5319, 2011.

[19] Q. Lin, "Technical analysis and stock return predictability: An aligned approach," *J. Financ. Mark.*, vol. 38, pp. 103–123, 2018.

[20] M. Thakur and D. Kumar, "A hybrid financial trading support system using multi-category classifiers and random forest," *Appl. Soft Comput.*, 2018.

[21] Z. Bitvai and T. Cohn, "Day trading profit maximization with multi-task learning and technical analysis," *Mach. Learn.*, vol. 101, no. 1–3, pp. 187–209, 2015.

[22] G. S. Atsalakis and K. P. Valavanis, "Surveying stock market forecasting techniques - Part II: Soft computing methods," *Expert Syst. Appl.*, vol. 36, no. 3 PART 2, pp. 5932–5941, 2009.

[23] M. Ballings, D. Van Den Poel, N. Hespeels, and R. Gryp, "Evaluating multiple classifiers for stock price direction prediction," *Expert Syst. Appl.*, vol. 42, no. 20, pp. 7046–7056, 2015.

[24] M. Gorenc Novak and D. Velušček, "Prediction of stock price movement based on daily high prices," *Quant. Financ.*, vol. 16, no. 5, pp. 793–826, 2016.

Puneet Misra is an Assistant Professor of Computer Science in the Department of Computer Science at the University of Lucknow, Lucknow, U.P., India. He received bachelor's degree (1995) in Physics and Math's and a dual master's degree in Electronics and Computer Applications, and a PhD degree (2003) from the University of Lucknow. He is currently engaged in research areas which include Soft computing, Artificial Intelligent Systems, human-computer interaction and issues related to cybercrime and its prevention policies etc.

Siddharth is a dual master's in computer science. He received his MTech degree from BITS Pilani, India, in System Software and MCA degree from BHU, Varanasi, India. Presently, he is pursuing his doctoral studies in the Department of Computer Science at University of Lucknow, India. He has more than 14 years of experience in the field of information technology. His research interests include artificial intelligence, machine learning and data mining for time series data. He is particularly interested in the field of machine learning and its application to the field of finance.