# Adaptive multi-column multi-stage machine learning pipeline for predicting stock price by solving a nonlinear optimization program

**Priyank Thakkar, Vijay Ukani, Yash Thesia, Vidhey Oza**
priyank.thakkar@nirmauni.ac.in, vijay.ukani@nirmauni.ac.in, 15BCE126@nirmauni.ac.in,
Institute of Technology, Nirma University, Ahmedabad, India
15BCE130@nirmauni.ac.in

**Article History**: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

**ABSTRACT**
Predicting a stock's next day price is a challenging task that researchers have long been attempting to address. Machine learning algorithms have shown to be successful at predicting next day price of a stock. A single machine learning model, however, suffers from limited stability and predictive power. Parallel models bring together a diverse set of learners to address these limitations. Predictions from individual diverse learners are typically merged into the final estimate using either a majority vote or simple or weighted averaging schemes. These schemes are simple and intuitive but static and sensitive to output from a single model that often leads to under-performance specifically for problems of a very dynamic nature, such as stock market prediction. To tackle these limitations, we present a new adaptive multi-column multi-stage pipeline, a novel technique to blend multiple individual learners by (1) calculating optimal column weights for the parallel models in the first-stage by solving a nonlinear minimization program (quadratic program) and (2) training a separate neural network based weight prediction module to estimate the optimal weights. We use Support Vector Regressiors (SVRs) as the multiple columns or multiple individual learners in the first-stage. We show the effectiveness of the proposed technique by attaining solid predictive performance on 8 stocks of National Stock Exchange, India. The proposed method outperforms other methods for 7 out of 8 stocks in the study.

**Keywords**—Machine Learning, Support Vector Regressor, Parallel Models, Stock Market, Quadratic Program

## 1. INTRODUCTION

Stock markets are regarded as the core of the world's economy involving billions of dollars of trade every day. The effective prediction of a stock's future price could generate substantial profit, and therefore researchers have been studying this problem for decades. The stock markets are dynamic, vary widely and have extremely non-linear character and complicated dimensionality, making stock market prediction a difficult problem [9, 13]. Most global clients, such as Goldman Sachs and The Global Investment Bank, expect India after the United States and China to be the third largest economy in the world. Global investors are increasingly interested in investing in Indian markets because of the advantage they can gain from exposure to emerging markets. These are markets that are growing at a high rate and offering opportunities that developed markets may not. These are ample reasons for studying Indian stock markets and predictions in these markets. National border security is one of the most important issue of any country due to intrusion, illegal immigration and terrorist threats. Modern surveillance and protection systems include border patrol vehicles, video surveillance system, control centers, permanent and mobile observation posts, and therefore incur high deployment and operational costs. Border areas are typically hostile, where it may be difficult to deploy required surveillance infrastructure. Traditional border surveillance and protection systems may not be sufficient to provide a complete border security.

In order to predict correctly in stock markets, different variables are taken into consideration. Stocks rely on several variables intrinsic in their activities, such as the state of the underlying economy, the recent performance of the company and the trust of shareholders. They are also governed by extrinsic variables such as recent changes in government policy, currency or commodity exchange rates, and industry-specific factors such as oil prices [5]. Fundamental analysis is a method of assessing securities by seeking to gauge the intrinsic value of a stock. Fundamental analysts study everything from the general conditions of the economy and industry to the financial condition and management of companies. Earnings, expenditures, assets, and liabilities are all significant features for fundamental analysts. Technical analysis differs from fundamental analysis in that the price and volume of the stock are the only inputs. The key hypothesis is that all known fundamentals are factored into the price, so there is no need to pay special attention to them. It utilizes distinct techniques and models to evaluate stock performance on the grounds of distinct criteria. Technical analysis has proven its prowess in both short- and long-term analysis. Machine learning models analyse past data to recognize future trends. Support Vector Regressor (SVR), Artificial Neural Networks (ANN) and Random Forest are among the most common models for predicting in stock markets [15, 16, 10]. A single machine learning model, however, suffers from limited stability and predictive performance.

In order to address these limitations, parallel models bring together diverse learners. Using either a majority vote or simple or weighted averaging schemes, estimates from individual diverse learners are typically merged into the final forecast. These schemes are straightforward and intuitive but static and susceptible to output from a single model, often contributing to under-performance specifically for very dynamic problems such as stock market prediction. To address these limitations, we are presenting a new adaptive multi-column multi-stage pipeline to blend individual learners by (1) calculating optimal model weights by solving a nonlinear optimization program (2) training a distinct neural network to estimate optimal weights.

The rest of the paper is structured as follows: Section 2 presents the related work in the field followed by the discussion of preliminaries in Section 3. The proposed approach with implementation details are described in Section 4, and results are discussed and analysed in Section 5. Section 6 ends with concluding remarks and potential future work.

## 2. RELATED WORK

Research on stock market prediction has been going on for years now, and researchers are constantly designing new algorithms to improve performance. In this section, we explore the literature on stock market prediction using machine learning methods. Approaches that involve multi-stage models are also discussed.

Cavalcante et al. carefully examined technical indicators and identified their usefulness in stock market prediction [3]. These indicators are designed by experts in the financial domain, such that each indicator is unique and has its own significance in technical analysis. These indicators have been used by many researchers to predict the next-day price or movement of stock or stock index. Patel et al. proposed the idea of trend-deterministic data and used them to predict trends in Indian stocks and stock indices [15]. Their core idea was to use the knowledge of the technical indicators, create trend signals independently from each indicator and use them to forecast the movement of the stock price. Technical indicators are of distinct types, such as momentum oscillators and volatility indicators, each representing a distinct aspect of the stock price. The selection of technical indicators is therefore essential for the optimal performance of any predictive model. This is discussed in detail in the study by [2].

Recently, some deep learning approaches have been proposed to address the task of stock market prediction. The authors in [17] developed ANN models for predicting stock market crashes by labeling data based on market signals to determine the best time to buy, sell or not trade the stock at all. LSTM has outperformed conventional approaches in a number of recent studies when it comes to predicting trends in time series data. In one specific instance, LSTM based language model was used to forecast market sentiment and better understand its future [4]. Feng et al. [7] used adversarial training to make the task of price movement prediction more robust by learning to ignore noise in data. A 3D CNN for feature extraction, and an LSTM network for prediction analysis were successfully combined in a study done in [19].

Despite the promise of deep learning methods, researchers are still using machine learning models coupled with novel ideas to boost prediction performance. One of the most significant elements affecting the efficiency of the machine learning based prediction model is the sort of model used for prediction. Perusing through the related work, we found that SVRs and SVMs are among the most popular choices for prediction on stock markets. Research on stock market prediction using these models was conducted in [20, 11, 8]. A multi-kernel SVR approach was presented in [20] as a two-stage forecast model for next-day price prediction. They effectively asserted that benefits from distinct combinations of hyper-parameters can be clubbed together to improve efficiency over single-kernel SVR or feedforward neural networks. The authors in [8] proposed a novel SVR variant called Adaptive SVR, intended to operate on high-frequency data. They proposed using particle swarm optimization to dynamically adjust the SVR hyper-parameters, and their results on 5-minute and 30-minute data were very promising. Research carried out in [11] is also a representative of the growing preference of research toward multi-column and multi-stage algorithms. The concept of using a weighted average on various models was employed in [21]. The authors suggested combining an SVR model with an ANN model using a weighted average whose weights were determined using the PSO algorithm. The researchers in [6] made a laudable comparison of conventional ensemble approaches. Algorithms such as bagging, random subspace and stacking were studied, and the pros and cons of all of them were discussed. The investigators in [12] suggested a neural network ensemble of 9 models, each trained on annualized returns from separate industrial sectors. This strategy was presented as a promising alternative to the ARMA model. The prowess of the ensemble approach was also tested in [14]. The authors learned 30 different neural networks on different sub-sets of technical indicators and identified 5 best performers using validation. These models were then used in the Brazilian stock market to generate market strategies. Their extensive experimentation is an evidence of the learning ability of the ensemble.

These proposals confidently demonstrate the effectiveness of multi-stage multicolumn methods, but they are static and non-adaptive in their merging scheme. The stock market is dynamic and involves different influencing situations at distinct times that require a lot of adaptation. In this paper, we propose an adaptive multi-column multi-stage machine learning pipeline for predicting stock price by solving (i) a nonlinear minimization program (quadratic program) and (ii) training a neural network based optimal weight prediction module. The proposed method addresses the limitations of the current multi-stage methods and makes the proposed pipeline adaptive to the different scenarios typically experienced in a stock market.

## 3. PRILIMINARIES

In this section, we discuss the basic knowledge of technical indicators required, followed by a preliminary discussion of models used in our implementations.

### 3.1 Technical Indicators

Technical indicators are statistical formulae generated through scientific analysis by different researchers in the financial markets sector. Each technical indicator has its own unique interpretation about the trend or volatility of a particular stock based on values of opening and closing price along with traded volume, high and low price. They can be broadly categorized into smoothing averages, momentum oscillators, trend, volatility, volume and overbought/oversold signal generators. The smoothing techniques used in general statistical analysis like Simple Moving Average (SMA) and Weighted Moving Average (WMA) are used to reduce the noise of the high-frequency data to detect the upward-downward or straight trends that would otherwise be invisible. Pure momentum indicators like ROC or PMO are used to gauge the rate of change of the trend. Advanced momentum oscillators like MACD use the difference of 2 moving averages of different windows to indicate new uptrend or downtrend away from the slower moving average. Other momentum indicators include RSI and MFI. RSI also generates overbought/oversold signals. If the value of the indicator rises above or falls below a certain threshold, the stock has likely experienced an overestimation or underestimation of the stock price and will soon correct itself. William's %R, CCI and the Stochastic oscillators (%K and %D) also generate such signals by making use of different movement patterns of the stock price.

Trend signals are generated by indicators like Aroon Oscillator as well as RSI. ADX is a trend indicator with 2 components: a True Range line and 2 Directional Movement lines. Ichimoku clouds are a unique trend signal generator, which use the concept of support and resistance clouds. Volume also plays a significant role in indications like how popular a trend shift is, or how much volume is required to generate that shift, though they are used generally as confirmation signals with stronger and more robust indicators. Ease of Movement (EMV) is an indicator that shows how much volume it required to achieve the price change that occurred in the day. ADL, OBV and CMF are some more volume indicators. Volatility indicators are used to gauge the volatility in the price movements. If the volatility is too high, the traders must be cautious of whipsaw signals given by other indicators. Bollinger Bands is one of the most famous volatility indicators that consists of 3 lines forming a 'band'. If the band widens, the standard deviation is high, implying high volatility. Mass Index and ATR use their own interpretations to judge the volatility of a stock's price.

A thorough understanding of these indicator categories is required to validate the idea behind using multiple models and applying ensemble learning to them. We use two SVRs in our proposed approach, and both models undergo specialized learning because they use different subsets of these indicators as inputs. This means that each model has its own representation of the problem and, as a result, its own point of view. The optimal integration of these models can certainly boost performance. Due to space constraints, we have not included formulas for calculating these indicators in this paper, but they can be found in any standard literature.

### 3.2 Support Vector Regressor (SVR)

SVR is a variation of Support Vector Machine (SVM). It is one of the most powerful machine learning techniques proposed in [18]. Due to the fact that SVR deals with continuous distribution, SVR has a challenging task to solve **than** SVM. It works on the idea of support vectors, which are data points that fall within a certain threshold distance from the hyperplane that is to be fit. The fitting takes place by only considering these support vectors and not the entire set of data points, hence dramatically improving performance. In other words, SVR sets the margin of tolerance E for approximation. This is a minimization problem applied to N dimension vector through fitting the hyperplane given in Eq. 1. The function $f(x, w)$ is the kernel function used for the algorithm.

$$Min. R = \frac{1}{2}\left||w|\right|^2 + c\left(\sum_{i=1}^{m} |y - f(x,w)|_{\varepsilon}\right)$$
$$OR$$
$$Min. R = \frac{1}{2}\left||w|\right|^2 + c\sum_{i=1}^{m}(\xi + \xi^*) \tag{1}$$
$$s.t. (w^T x_i + b) - y_i \leq \varepsilon - \xi_i$$
$$y_i - (w^T x_i + b) \leq \varepsilon - \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0, i = 1,2,3, \dots, m$$

## 4. PROPOSED APPROACH AND IMPLEMENTATION DETAILS

In this section, we establish the idea behind our approach, and then move on to details of our adaptive multi-column two-stage ensemble approach: SVR-QPP-ANN.

### 4.1 Data Preparation

We implemented our approaches on 8 stocks from National Stock Exchange (NSE), India. These stocks include ASIANPAINT, BPCL, BRITANNIA, HDFCBANK, HINDUNILVR, MARUTI, RELIANCE, and ZEEL. The essential data of these stocks were retrieved using the Alpha Vantage API [www.alphavantage.co]. The selection of these stocks was done in a manner to include industries from different sectors. We fetched data of recent 11 years. The dataset consisted of opening and closing price, high and low price of the day, and the traded volume of the stock. It also included the adjusted closing price that includes adjustments made due to market events like stock split or dividends issued. Using Phase AdjustPrices of Algorithm 1, we adjusted the opening, high and low prices, so that comparisons can be made on a common ground. Technical indicators were then derived from these datasets. An optimal window size of 10 for most of the indicators was chosen based on an exhaustive empirical study. Various values of window size used in the study can be seen in Table 1. The technical indicators used in the implementation are discussed abstractly in Section 3.1. 85% of this dataset was then used for training all the models.

Since normalization is recommended for better performance in any machine learning model, we used Z-score normalization (shown in Eq. 2) to scale the data. This method uses mean and standard deviation to bring all the data to a common scale.

$$X_i^{norm} = \frac{X_i - \mu_X}{\sigma_X} \tag{2}$$

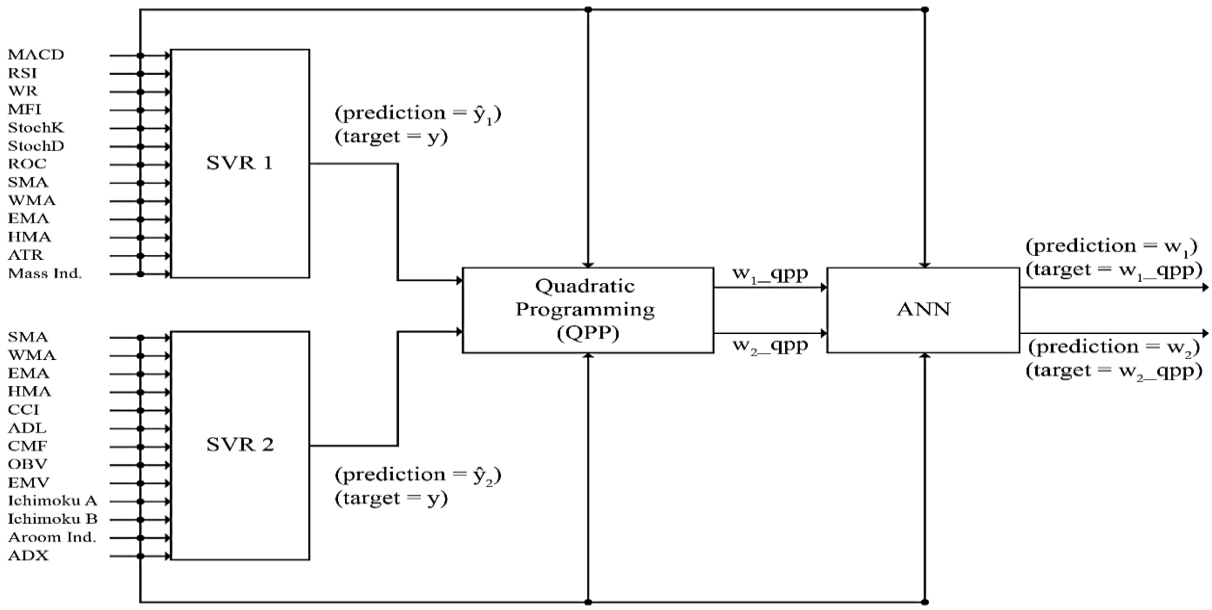


**Figure 1.** Proposed Multi-Column Multi-Stage Approach

### 4.2 Proposed Multi-Column Multi-Stage Prediction Pipeline (SVR-QPP-ANN)

In a typical regression-based ensemble scenario, the second-stage uses predictions from individual models and decides the outcome based on averaging or majority vote strategy. Also, it is generally observed that since the first-stage individual models are trained using different knowledge bases, they have their own interpretation of the domain. This is true regardless of the fact that their task is the same: prediction of next-day stock price. There are many strategies to integrate predictions of these first-stage models. The simplest strategy is to generate a simple mean over the outputs of the individual. Though it is perfectly logical, it completely ignores the separate learned capabilities of the individual models and gives each model an equal footing in the ensemble. The next rational idea is to perform a weighted average, and hence use the support of the models that already performed better than others. The main challenge here is to find the optimal weights. Our approach finds these weights by a second machine learning model. The idea is that rather than finding optimal weights by using economic or financial rules, we let a machine learning model understand the importance of each model output on its own, and hence let it generate these weights on its own.

Hence, we propose an ensemble pipeline that implements this idea, which we call SVR+QPP+ANN. In the proposed approach, the first-stage models are SVR models that are optimized for their own sets of knowledge bases. The first-stage consists of 2 RBF-kernel SVR models, called SVR1 and SVR2. This choice of using SVR as prediction models was done in line with results from many researches [20, 8, 11]. The inputs are various indicators as displayed in Figure 1, and the output is the predicted next-day stock price. SVR1 was trained using momentum-based and volume-based indicators. Momentum oscillators are used to generate signals on the rate of change in closing prices. Volume indicators are generally used as support for other indicators. Combined, these indicators can predict next-day stock price with reinforced strength. SVR2 was learned on trend-based and volatility indicators along with CCI, an overbought/oversold signal generator. Trend indicators determine whether a change in price signals towards a major shift in stock price trend. Volatility indicators are used as warnings to potential whipsaw signals, i.e. signals that quickly reverse their opinion. Together with CCI, these indicators generate a balanced prediction of the next-day stock price. Hyper-parameters were tuned from the set given in Table 1. The final values of $c_r$ and $\gamma$ used were 5 and 0.001 respectively for both models.

**Table 1.** Hyper-parameter values used for Grid Search

| **Parameter** | **Grid used** |
|---|---|
| Indicator Window size (w) | 10, 15, 20, 25, 30, 50 |
| Gamma in kernel function ($\gamma$) | 0.001, 0.01, 0.1, 1, 10 |
| Regularization parameter ($c_r$) | 0.01, 0.1, 1, 2, 3, 4, 5 |

**Algorithm 1.** Proposed multi-column two-stage prediction pipeline.

**Phase AdjustPrices:**
dataset.adjust_ratio ← (dataset.adjusted_close)/(dataset.close)
**for** each col in [dataset.open, dataset.low, dataset.high] **do**
    col ← col ∗ dataset.adjust_ratio
**end for**
**Phase DataProcessing:**
indicators_dataset ← data_to_indicators(dataset)
normalized_dataset ← normalize(indicators_dataset)
subset1 ← normalized_dataset.get_subset1()
subset2 ← normalized_dataset.get_subset2()
**Phase SVRTraining:**
svr1.train(in = (subset1, $c_r$, $\gamma$), target = y)
svr2:train(in = (subset2, $c_r$, $\gamma$); target = y)
$\hat{y}1$ ← svr1.predict(subset1)
$\hat{y}2$ ← svr2.predict(subset2)
**Phase EnsembleTraining:**
w1_qpp, w2_qpp ← qpp.minimize(subset1, subset2, $\hat{y}1$, $\hat{y}2$, y)
ann.train(in = [subset1, subset2], target = [w1_qpp, w2_qpp])
**Phase OptimalWeightPrediction:**
w1, w2 ← ann.predict(in = [subset1, subset2])
**Phase FinalPrediction:**
Final_Prediction = w1 ∗ $\hat{y}1$ + w2 ∗ $\hat{y}2$

Once the SVRs in the first stage have been trained, we create a new training set that pairs input technical indicators with optimal column weights for the second stage. For each input data point, a vector = [technical_indicators_subset1, ..., technical_indicator_subsetC] is constructed by concatenating the inputs of each SVR, where C is the number of columns (It is worth noting here that the approach can be generalized to C models in the first stage, where C can be any positive integer greater than 1). We used C = 2 in our implementation.

Furthermore, for each input data point, the output of each column is collected into a matrix $\hat{Y} = [\widehat{y1}, \ldots, \widehat{yC}] \in R^{D\,X\,C}$, with each column being the output of one of the SVR columns, $\hat{y}_c$. We perform the following non-linear minimization (quadratic program) to determine the ideal linear weighting of the SVR columns for that given input data point.

$$
\begin{aligned}
&minimize_{\{w_c\}} \frac{1}{2} ||\hat{Y}_w - y||^2 \\
&subject\ to\ 0 \le w_c \le 1 \quad \forall c \\
&1 - \delta \le \sum_{c=1}^{C} w_c \le 1 + \delta \\
&Final\_Prediction = \sum_{i=1}^{C} w_c \times \hat{y}_c
\end{aligned}
\tag{3}
$$

In this case, $w \in R^C$ is the vector of weights $w_c$ that corresponds to each SVR column c. Weights restricted between 0 and 1 were shown to improve weight predictions by reducing overfitting. The constraint in Eq. 3 aids in the avoidance of degenerate cases.

We now train an ANN with this set of optimal weights, as shown in Fig. 1. Using the optimal weight training set described above, ANN was trained to take the feature vector as input and produce a weight vector w. Algorithm 1 summarizes the learning pipeline structure for the proposed multi-column multi-stage pipeline known as SVR+QPP+ANN. The proposed method is based on the work published in [1] for robust image denoising using multiple stacked sparse denoising autoencoders (SSDA).

## 5. RESULTS & DISCUSSIONS

We compare the proposed SVR+QPP+ANN approach to individual SVR models (SVR1 and SVR2), an SVR trained on all technical indicators, the mean of SVRs, the weighted average of SVRs, and SVR+MLR and SVR+SVR. When compared to the proposed approach, the last two approaches used multiple linear regression (MLR) and SVR in the second stage instead of QPP+ANN. We use root mean squared error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) metrics for the comparison. However, due to space constraints, we have shown only MAPE values in Table 2. MAPE can be calculated as shown in Eq. 4.

$$
MAPE = \sum_{i=1}^{n} \frac{|y_i - \hat{y}_i|}{|y_i|} * 100 \tag{4}
$$

The first observation is the generally better performance of at least one of SVR1 or SVR2 as compared to the Full SVR. This asserts the idea that individual models given a fraction of the knowledge base perform better than a model that is trained on the whole base. As a result, even the simple Avg. SVR works better than Full SVR. The next observation is the generally similar performance of Avg. SVR and Weighted Avg. SVR. For comparison, we consider Table 3 here. MAPE is a percentage error metric, meaning MAPE of 1 indicates error of 0.01. Since the MAPE values of both these models are same for up to 2 decimal places, the difference between the models is less than 0.01% in general. In most cases, the proposed approach outperforms these simple ensemble techniques as well as SVR+MLR and SVR+SVR.

**Table 2.** Mean Absolute Percentage Error (MAPE)

| Stock | SVR1 | SVR2 | Full SVR | Avg. SVR | Wtd. Avg. SVR | SVR + MLR | SVR + SVR | SVR + QPP + ANN |
|---|---|---|---|---|---|---|---|---|
| BPCL | 5.05 | 3.94 | 3.88 | 4.44 | 4.44 | 2.76 | 3.16 | **1.89** |
| ASIANPAINT | 3.46 | 2.74 | 2.02 | 2.83 | 2.83 | 2.47 | **1.65** | 1.79 |
| BRITANIA | 4.62 | 4.35 | 3.85 | 4.3 | 4.3 | 3.32 | 3.19 | **2.15** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| HDFCBANK | 4.22 | 4.45 | 4.06 | 4.24 | 4.24 | 2.49 | 2.85 | **2.28** |
| HINDUNILVR | 3.0 | 3.11 | 2.39 | 2.89 | 2.89 | 2.94 | 1.9 | **1.45** |
| MARUTI | 6.4 | 7.76 | 6.87 | 7.05 | 7.05 | 5.92 | 5.22 | **4.5** |
| RELIANCE | 5.38 | 6.42 | 5.7 | 5.89 | 5.88 | 5.27 | 5.43 | **4.74** |
| ZEEL | 2.73 | 2.49 | 2.11 | 2.51 | 2.51 | 2.02 | 1.88 | **1.77** |

This clearly shows the superiority of the proposed method which integrates first stage models using optimal weights as estimated through quadratic programming and artificial neural network.

## 6. CONCLUSION & FUTURE WORK

The paper proposes multi-column multi-stage pipeline for predicting next day's closing price of a stock. Ensemble or multi-stage approaches stem from the idea of multiple models being earned on different subsets of the knowledge base. When these individual models are combined together, the ensemble performs better than a single model that is learned on the entire knowledge base. The proposed approach is one step ahead of the general simple or weighted averaging pipeline. The approach employs quadratic programming and artificial neural network following the first-stage models to learn optimal weights for these models. Experiments were performed on 8 Indian stocks. The results were compared with different averaging ensemble models, Full SVR, SVR+MLR and SVR+SVR. The proposed approach outperformed all the models for 7 out of 8 stocks used in the study.

In future, we can improve this approach in two ways: (1) adding more knowledge base for the first-stage models like market sentiments, oil prices, related stocks or index giving more promising representation; (2) improving the ensemble model by making a system that predicts the results not only using inputs and outputs of the first-stage models but also using their latent vectors.

## REFERENCES

[1] Agostinelli, F., Anderson, M.R., Lee, H.: Adaptive multi-column deep neural networks with application to robust image denoising. In: Advances in Neural Information Processing Systems, pp. 1493{1501 (2013)

[2] Atsalakis, G.S., Valavanis, K.P.: Surveying stock market forecasting techniques {part ii: Soft computing methods. Expert Systems with Applications 36(3), 5932{5941 (2009)

[3] Cavalcante, R.C., Brasileiro, R.C., Souza, V.L., Nobrega, J.P., Oliveira, A.L.: Computational intelligence and financial markets: A survey and future directions. Expert Systems with Applications 55, 194{211 (2016)

[4] Chang, Q.: The sentiments of open financial information, public mood and stock returns: an empirical study on Chinese growth enterprise market. International Journal of Computational Science and Engineering 23(2), 103{114 (2020)

[5] Chen, N.F., Roll, R., Ross, S.A.: Economic forces and the stock market. Journal of business pp. 383{403 (1986)

[6] Cheng, C., Xu, W., Wang, J.: A comparison of ensemble methods in financial market prediction. 2012 Fifth International Joint Conference on Computational Sciences and Optimization pp. 755{759 (2012)

[7] Feng, F., Chen, H., He, X., Ding, J., Sun, M., Chua, T.S.: Enhancing stock movement prediction with adversarial training. In: IJCAI (2019)

[8] Guo, Y., Han, S., Shen, C., Li, Y., Yin, X., Bai, Y.: An adaptive svr for high-frequency stock price forecasting. IEEE Access 6, 11397{11404 (2018)

[9] Guresen, E., Kayakutlu, G., Daim, T.U.: Using artificial neural network models in stock market index prediction. Expert Systems with Applications 38(8), 10389{10397 (2011)

[10] Henrique, B.M., Sobreiro, V.A., Kimura, H.: Stock price prediction using support vector regression on daily and up to the minute prices. The Journal of finance and data science 4(3), 183{201 (2018)

[11] Kao, L.J., Chiu, C.C., Lu, C.J., Yang, J.L.: Integration of nonlinear independent component analysis and support vector regression for stock price forecasting. Neurocomputing 99, 534{542 (2013)

[12] Lahmiri, S.: Ensemble with radial basis function neural networks for Casablanca stock market returns prediction. 2014 Second World Conference on Complex Systems (WCCS) pp. 469{474 (2014)

[13] Lee, T.S., Chiu, C.C.: Neural network forecasting of an opening cash price index. International Journal of Systems Science 33(3), 229{237 (2002)

[14] de Mello Assis, J., Pereira, A., Couto e Silva, R.: Designing financial strategies based on artificial neural networks ensembles for stock markets pp. 1{8 (2018). DOI 10.1109/IJCNN.2018.8489688

[15] Patel, J., Shah, S., Thakkar, P., Kotecha, K.: Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Systems with Applications 42(1), 259{268 (2015)

[16] Patel, J., Shah, S., Thakkar, P., Kotecha, K.: Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Systems with Applications 42(1), 259{268 (2015)

[17] Tabar, S., Sharma, S., Volkman, D.: A new method for predicting stock market crashes using classification and artificial neural networks. International Journal of Business and Data Analytics 1(3), 203{217 (2020)

[18] Vapnik, V.N.: An overview of statistical learning theory. IEEE transactions on neural networks 10(5), 988{999 (1999)

[19] Yang, C., Zhai, J., Tao, G.: Deep learning for price movement prediction using convolutional neural network and long short-term memory. Mathematical Problems in Engineering 2020 (2020)

[20] Yeh, C.Y., Huang, C.W., Lee, S.J.: A multiple-kernel support vector regression approach for stock market price forecasting. Expert Systems with Applications 38(3), 2177{2186 (2011)

[21] Zheng, Q., Chen, Y., Liu, Z.: Ensemble model of intelligent paradigms for stock market forecasting. First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008) pp. 205{208 (2008)