Robust Time Series Forecasting via Time Differencing and Stacking

Aditya Bobde
Dept. of Artificial Intelligence
G H Raisoni College of Engineering
Nagpur, India
bobde_aditya.ai@ghrce.raisoni.net

Vishal Narnaware
Dept. of Artificial Intelligence
G H Raisoni College of Engineering
Nagpur, India
narnaware_vishal.ai@ghrce.raisoni.net

Shivam Tawari
Dept. of Artificial Intelligence
G H Raisoni College of Engineering
Nagpur, India
tawari_shivam.ai@ghrce.raisoni.net

Achamma Thomas
Dept. of Artificial Intelligence
G H Raisoni College of Engineering
Nagpur, India
achamma.thomas@raisoni.net

***Abstract*—Future price prediction of stocks is a challenging task for researchers. Artificial intelligence and enhanced computing powers have shown to be successful in forecasting price of the stock. With the help of this study, we propose machine learning strategies that can be incorporated with stock market systems to help users get the stock's closing price the next day. In this study, the performance of the most popular and widely applied linear based models are compared like - Linear Regression, Polynomial, SGDRegression and Support Vector Machine, Regularized Models like - Lasso, Ridge and Elastic-Net Regression, and Robust Models like - TheilSen and RANSACRegressor on daily stock price data fetched from Zerodha with technical analysis as additional features and pre-processed with time differencing. We propose a novel architecture to robustly forecast time series data by stacking SGDRegressor, Polynomial Regressor and Support Vector Machine with meta model as Support Vector Machine with RBF kernel.**

***Index Terms*—machine learning, linear models, linear regression, polynomial regression, support vector machine, sgdregressor, theilsen regression, stacking, stationarity, stock market, time series**

I. Introduction

The forecasting of the stock market is an important financial aspect that has recently piqued the interest of traders and developers. It involves creating a relation between the past performance of a stock to predict its future. Forecasting stock prices is an essential part of Algorithmic Trading. Algorithmic trading has gained popularity in the fields of artificial intelligence and finance in recent years.

Making intelligent machines is the main goal of AI. Day trading is dangerous in view of the transient conduct of business sectors that reflect billions of quickly fluctuating qualities receptive to developing conditions that rough an arbitrary walk. More than 95% of traders lose money. It requires a great deal of consideration and affectability on the market.

A system with rapid web associations and great power can execute a huge number of exchanges during a day. This is called high-frequency trading. No human can rival these calculations, they're incredibly quick and more exact. The most effective way to represent such processes is with machine learning. It improves accuracy by predicting a market value that is close to the tangible value.

This work's main objective is to compare methods for forecasting future price changes in financial markets based on historical results. The fundamental premise is that the efficient market hypothesis is false and that, similarly to the reasoning behind technical analysis of stock price charts, the past may hold some clues to the future that can be gleaned by statistical methods.

In other words, it is assumed that certain patterns in financial markets repeat themselves, allowing for the use of historical data to forecast price changes in the future.

II. Related Work

The AutoRegressive (AR) model and the AutoRegressive Integrated Moving Average (ARIMA) model are two examples of Box-Jenkins models [1] that were used to address the stock market's intrinsic non-linearity and non-stationarity. These models, on the other hand, are based on the assumption that the time series under consideration are linear and stationary.

Osman Hegazy proposed an algorithm in [2] that combines Particle Swarm Optimization (PSO) and the Least Square Support Vector Machine (LS-SVM). It is used to optimise LS-SVM in order to forecast the daily price of a stock. To improve forecast accuracy and lower the likelihood of over-fitting as well as local minima problems, it chooses the best free parameter combination for LS-SVM. Similarly, [3] proposed a new prediction algorithm that uses the temporal relationship between global stock markets and a variety of financial products to forecast the next-day pattern in the stock market using Support Vector Machines (SVM).

The authors of [4] have demonstrated the efficacy of a hybrid strategy, increasing the amount of time delays and network architecture components using genetic algorithms (GA) and neural networks (NN). Genetic Algorithms (GA) and Neural Networks (NN) to augment the number of time delays and network architectural elements, improving the effectiveness of building Artificial Neural Network (ANN) models. A hybrid ARIMA and SVM model that leverages the inherent strengths of both models in anticipating stock price problems was suggested in [5].

Stock market data are complex, non-stationary and chaotic. Principal Component Analysis (PCA) is used to discover the most important features in predicting stock price. Applying PCA on stock data with Technical Analysis (TA) and further passing it to SVM and ANN model helps in decreasing the noise within features [6]. A combination of the TA and Fundamental Analysis (FA) has also been explored with a hybrid model using both TA and FA as feature variables of stock market indicators and ANN as predictor model [7].

A study was conducted for evaluating Financial distress using Machine Learning. The study concluded with proposing SVM model based on the Gaussian RBF kernel for evaluating Financial distress [8]. They have also used Back Propagation Neural Network (BPNN) and then showed the comparative study between SVM and BPN Network. In [9], authors have put forward the concept of Reinforcement Learning for trading in the stock market. Additionally, they have created a new reinforcement learning framework that enables the creation of successful trading instruments using a smaller set M_0 of market experiences.

III. METHODOLOGY

A. Technical Indicators

- Relative Strength Index

  The Relative Strength Index (RSI) [10] is a momentum oscillator that has a bound range of 0 and 100. The RSI determines recent price movements' magnitude in order to pinpoint overvalued and undervalued regions. The RSI emits signals that indicate whether the price of an asset is more likely to rise or fall.

  There are three components of RSI: Average Gain, Average Loss and Relative Strength [11]. 14 periods is commonly used for RSI calculation [10].

$$RSI = 100 - 100/(1 + \mathit{RelativeStrength}) \quad (1)$$

$$RelativeStrength = (\mathit{Avg\ Gain})/(\mathit{Avg\ Loss}) \quad (2)$$

  The first calculation of Average Gain and Average Loss are 14 period averages. The subsequent calculations also take into account the current gain or loss as well as the previous averages.

- Moving Average Convergence Divergence

  The momentum indicator known as moving average convergence divergence (MACD) tracks trends and depicts the relation between the two moving averages of a stock's price. It aids in determining whether the market sentiment is bullish or bearish in terms of price. It is computed by deducting the 12-period exponential moving average from the 26-period exponential moving average.

$$MACD = 12PeriodEMA - 26PeriodEMA \quad (3)$$

  There are two major components of MACD: MACD line and signal line. When the MACD line crosses from underneath to over the signal line then it is considered as bullish elsewise bearish.

- Bollinger Bands

  Bollinger Bands are a price range shown both above and below a simple average line at one standard deviation level. Its three lines are a simple rolling average (middle band), a higher band, and a bottom band. To measure how spread out the price is from normal esteem we use standard deviation. A price's relative highs and lows are indicated by the Bollinger Bands. Price is defined as high at the Upper Bollinger Band (UBB) and low at the Lower Bollinger Band (LBB).

  The Bollinger Bands indicator is based on volatility and standard deviation. One standard deviation, two standard deviations, and three standard deviations are all within which 68.26%, 95.44%, and 99.73% of the values, respectively, are contained. Bollinger Bands have a straightforward premise: 90% of the price fluctuation happens within the two zones. A major signal is any breakdown below or any breakout above.

  The default value for plotting of standard deviations has been set to two [12].

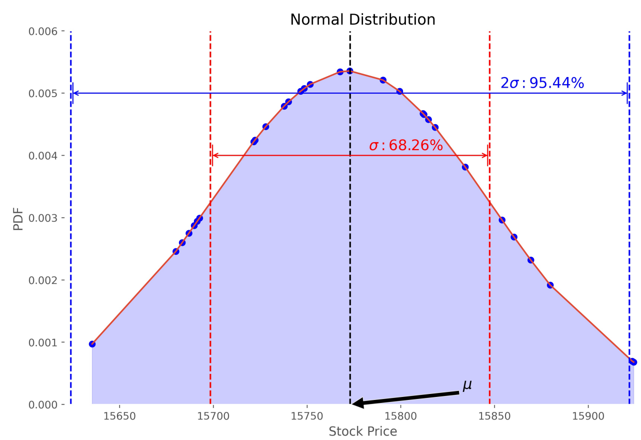

Fig. 1. Normal Distribution and Standard Deviation. X-axis consists of the stock prices in the past 30 days, PDF has been used to generate corresponding y values.

$$\sigma = \sqrt{\frac{\sum_{j=1}^{N}(X_j - \bar{X})}{N}} \quad (4)$$

$$Where, \bar{X} = \frac{\sum_{j=1}^{N} X_j}{N} \quad (5)$$

$$And, UpperBollingerBand(UBB) = \bar{X} + 2\sigma \quad (6)$$

$$MiddleBollingerBand(MBB) = \bar{X} \quad (7)$$

$$LowerBollingerBand(LBB) = \bar{X} - 2\sigma \quad (8)$$

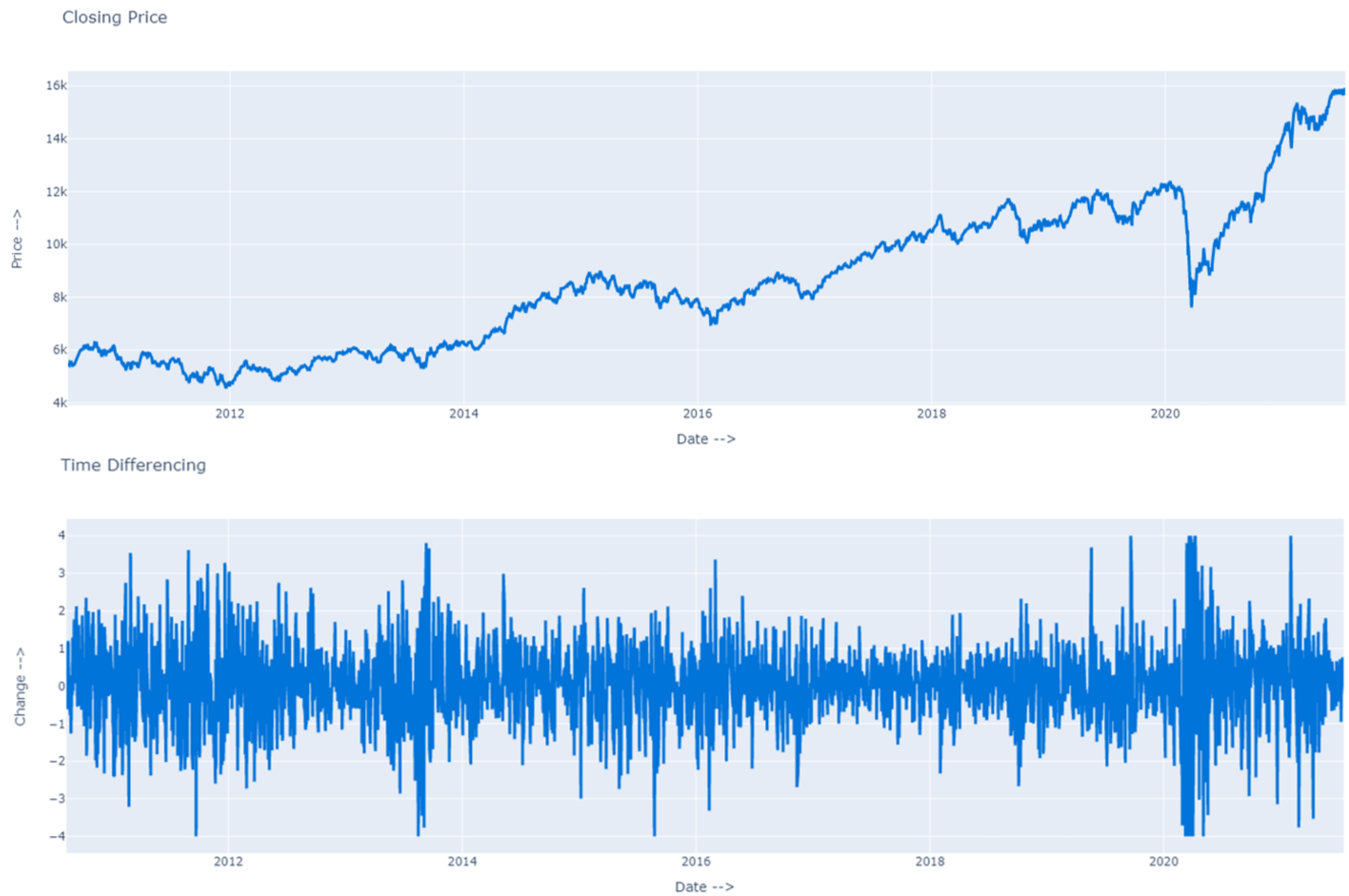

Fig. 2. Difference between Stationary and Non-Stationary wave. a) Upper figure shows the price in uptrend and; b) lower figure shows the same price but time differenced and in percentage change.

B. Dataset

The authors of this paper have performed predictions on the NIFTY50 index. The dataset was called from kite Zerodha's API [13] of past 2700 days (from 13 August 2010 to 16 July 2021). The data initially consisted of date, open, high, low and close columns.

Let X_t be a time series wave from which data is sampled at a constant interval, m_t the trend in data, s_t the seasonality and Y_t noise in data. Then the time series wave can be written as:

$$X_t = m_t + s_t + Y_t \tag{9}$$

While predicting time series, ML models expect data to be stationary. In other words, trend and seasonality should not be present. By removing changes in the time series' level, differencing may assist in stabilising the mean of a time series. and reduce (or eliminate) trend and seasonality. The Backward Shift Operator outputs the previous observation in time. The Backward Shift Operator is defined for a time series X_t as

$$BX_t = X_{t-1} \tag{10}$$

The Lag-1 Difference Operator outputs the difference between current data point in time with previous data point.

$$\triangledown X_t = X_t - X_{t-1} = (1 - B)X_t \tag{11}$$

Time differencing (Lag-1 Differencing) is used in order to convert increasing or bullish index (data) into stationary waveforms. We have further converted time differenced wave to percentage change in price (e.g., percent change in today's high with respect to yesterday's high).

$$\%change = \frac{\triangledown X_t}{X_t} * 100 \tag{12}$$

"Figure 2" shows the difference between a stationary and non-stationary wave. The closing price of NIFTY50 index is shown in plot 2a and time differenced stationary data in plot 2b.

Technical Analysis is also added in features as discussed in section III(A) of this paper. RSI and MACD are range bound indicators and therefore we have skipped the time differencing part on these and directly added lags as features. Majority of the candlestick patterns (e.g., pin bar, hammer, shooting/morning star, three crow/soldiers) are found within 3 candles and hence lags are added up to 3 days. A lag of 6th day is also added in order for the model to get an understanding

TABLE I
DATA FEATURES

Feature	Description
%Open	Percentage change in today's open with respect to yesterday's open price.
%High	Percentage change in today's high with respect to yesterday's high price.
%Low	Percentage change in today's low with respect to yesterday's low price.
%Close	Percentage change in today's close with respect to yesterday's close price.
%Body	Percentage change in today's close with respect to today's open price.
%Candle	Percentage change in today's high with respect to today's low price.
RSI	Value of today's Relative Strength Index
MACD	Value of today's MACD
MACD Signal	Value of today's MACD Signal
%UBB	Percentage change in today's UBB with respect to yesterday's UBB.
%MBB	Percentage change in today's MBB with respect to yesterday's MBB.
%LBB	Percentage change in today's LBB with respect to yesterday's LBB.
%Open-1, %Open-2, %Open-3, %Open-6	1-day, 2-day, 3-day and 6-day lags of percent change in open price.
%High-1, %High-2, %High-3, %High-6	1-day, 2-day, 3-day and 6-day lags of percent change in high price.
%Low-1, %Low-2, %Low-3, %Low-6	1-day, 2-day, 3-day and 6-day lags of percent change in low price.
%Close-1, %Close-2, %Close-3, %Close-6	1-day, 2-day, 3-day and 6-day lags of percent change in close price.
%Body-1, %Body-2, %Body-3, %Body-6	1-day, 2-day, 3-day and 6-day lags of percent change in body.
%Candle-1, %Candle-2, %Candle-3, %Candle-6	1-day, 2-day, 3-day and 6-day lags of percent change in candle.
RSI-1, RSI-2, RSI-3, RSI-6	1-day, 2-day, 3-day and 6-day lags of RSI.
MACD-1, MACD-2, MACD-3, MACD-6	1-day, 2-day, 3-day and 6-day lags of MACD.
MACD Signal-1, MACD Signal-2, MACD Signal-3, MACD Signal-6	1-day, 2-day, 3-day and 6-day lags of RSI.
%UBB-1, %UBB-2, %UBB-3, %UBB-6	1-day, 2-day, 3-day and 6-day lags of percent change in UBB.
%MBB-1, %MBB-2, %MBB-3, %MBB-6	1-day, 2-day, 3-day and 6-day lags of percent change in MBB.
%LBB-1, %LBB-2, %LBB-3, %LBB-6	1-day, 2-day, 3-day and 6-day lags of percent change in LBB.

what happened last week. "TABLE I" summarizes the feature engineering part.

C. Machine Learning Models

- Linear Regression
  The linear regression approach involves fitting a linear equation to the observed data, to model the connection between two variables, the regression analysis approach is used. One of the variables is independent, whereas the other is dependent. It has equation of the form:

$$h_\theta(x) = \sum_{i=0}^{d} \theta_i x_i = \theta^T x \tag{13}$$

  Where θ is a parameterized vector that represents the space of linear functions mapping from X to Y. X represent the input value space, and Y signifies the set of possible output values. They hold the assumptions of Linear Regression. The first assumption is Linearity, X and Y have Linear relationship. Another is Homoscedasticity, where the residual variance is constant for any value of X. The third assumption is independence, which states that observations are not dependent on one another. And Normality, for any constant value of X, Y is distributed normally.
- Polynomial Regression
  A polynomial of nth degree in X is used in polynomial regression to attempt to express the connection between both the independent variable X as well as the dependent variable Y. It fits the Non-Linear relationship between X and the conditional mean of Y. It can be represented as:

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_n x^n \tag{14}$$

  Where n is defined as the degree of polynomial.
- Stochastic Gradient Descent Regression
  The Stochastic Gradient Descent (SGD) Regressor uses a regularized linear function and SGD learning to build an estimator. Essentially, it is a linear model that was fitted by using SGD to minimise a regularised empirical loss. A regularizer is a penalty (L1, L2, or Elastic Net) that is applied to the loss function in order to reduce the size of the model's parameters. To enable the learning of sparse models and online feature selection, the update is trimmed to 0.0 if the parameter update is greater than 0.0 as a result of the regularizer.
- Elastic-Net
  Elastic-Net is a regularized regression algorithm which combines both L1 and L2 penalty. It combines Lasso and Ridge Regression methods into one. The benefit of this combination is that it allows for the learning of a sparse model with few non-zero weights(from Lasso Model, L1 regularization), and this still maintains the regularization properties of Ridge method (L2 regularization). The objective function to minimize,

$$min \frac{1}{2n_m} ||X_W - Y||_2^2 + \alpha\rho||W||_1 + \frac{\alpha(1-\rho)}{2}||W||_2^2 \tag{15}$$

  Where α is a constant, that multiplies with the L1 and L2 term and decides the penalization of the model and m is number of samples.
- Support Vector Machine
  Support Vector Machines (SVM), developed by Vladimir Vapnik are based on Statistical Learning Theory [15]. SVM Regression has been previously used for Volatile Market Prediction [16]. Support Vector Machine Regression (SVR) goal is to find a linear hyperplane that matches the multidimensional input vectors to the output values. The outcome is then utilised to forecast future output values from a test set.

$$minimize_{w,b,\gamma}\frac{1}{2}w^Tw+T\sum_{i=1}^{n}\gamma^{(i)} \tag{16}$$

Subject to,

$$c^{(i)}(w^Tx^{(i)}+b)\leq 1-\gamma^{(i)} \tag{17}$$

and

$$\gamma^{(i)}\geq 0 \text{ for } i=1,2,3,...,n$$

- TheilSen Regression
  TheilSenRegressor is a Machine Learning algorithm that estimates the model by computing the slopes and intercepts of a subpopulation of all possible combinations of p subsample points. In order to fit an intercept, p must be bigger than or equal to n features + 1. The final slope and intercept are then calculated using the spatial median of these slopes and intercepts. TheilSenRegressor employs a generalisation of the median over many dimensions. It is hence immune to multivariate outliers.

D. Evaluation Metrics

- Mean / Bias
  Mean or Bias is the summation of difference between predicted and actual values. Stock Market data is generally biased on the bull side. Mean Error metric can be used to get an understanding of which side the model is, bullish or bearish, the preceding is preferred. The formula is given in equation 18

$$Mean=\frac{1}{m}\sum_{i=1}^{i=m}(\hat{y}_i-y_i) \tag{18}$$

  Or in vectorized form,

$$Mean=\frac{1}{m}(\hat{y}-y) \tag{19}$$

  Where, m is defined as the number of training examples, $\hat{y}$ and y are vectors of size m containing predicted and actual values respectively.
- Mean Absolute Error
  Mean Absolute Error (MAE) is an evaluation method for regression task. MAE is the average absolute difference between two vector elements. This metric is used to know how off are predictions from actual values. The MAE calculates the magnitude of errors for the whole group by averaging the absolute errors for a set of predictions and observations. MAE is calculated with the formula shown in equation 20.

$$MAE=\frac{1}{m}\sum_{i=1}^{i=m}|\hat{y}_i-y_i| \tag{20}$$

  Or,

$$MAE=\frac{1}{m}|\hat{y}-y| \tag{21}$$

  Where, m is defined as the number of training samples, $\hat{y}$ is another m sized vector that contains predictions from the model and y is another vector of size m whose elements are the actual values.
- Mean Squared Error
  The average of the difference in squared terms between the predicted and actual values is known as the mean squared error (MSE), often referred to as the mean squared deviation. The formula of MSE is similar to squared L2 Norm. MSE is always positive. Equation 22 shows MSE in mathematical form.

$$MSE=\frac{1}{m}\sum_{i=1}^{i=m}(\hat{y}_i-y_i)^2 \tag{22}$$

  Or,

$$MSE=\frac{1}{m}(\hat{y}-y)^2 \tag{23}$$

- Root Mean Squared Error
  Another regression metric is Root Mean Squared Error (RMSE) which is the standard deviation of the prediction error. RMSE is an estimate of how far the actual values are from the prediction or the line of best fit. Similar to MSE, RMSE is also always positive. RMSE can be calculated by Equation 24.

$$RMSE=\sqrt{\frac{1}{m}\sum_{i=1}^{i=m}(\hat{y}_i-y_i)^2} \tag{24}$$

  Vectorized form,

$$RMSE=\sqrt{\frac{1}{m}(\hat{y}-y)^2} \tag{25}$$

IV. Implementation and Results

The preprocessed and cleaned data with added features as summarized in "TABLE I", is split into two parts – train and test. We reserved 5% (103 observation / days) of the original dataset for testing and evaluation of metrics presented in this paper. The size of the train data is small and therefore we have used Cross Validation (CV) instead of further splitting the data into train and validation.

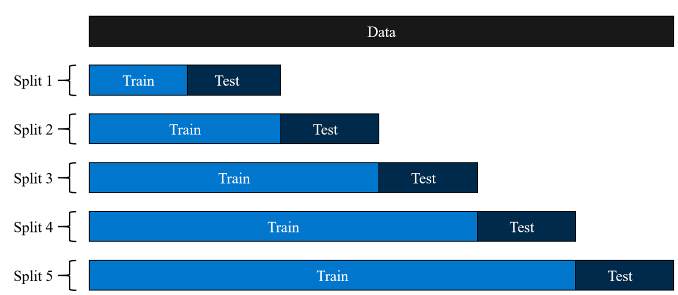

Fig. 3. Hold-out Cross Validation for Time Series Data, split is incremental.

With the time series data, we used hold-out CV, where a subset of data, divided temporally, is reserved for testing the model performance in order to accurately imitate "real world forecasting environment" and prevent data leakage. Figure 3 shows the split of train and validation data under CV. In this

TABLE II
EVALUATION METRICS AND ML MODELS

ML Models	Error Metrics							
	Mean/Bias		MAE		MSE		RMSE	
	Mean	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*
Linear Regression	0.0002	0.1093	0.7830	0.2803	1.6734	1.9220	1.1507	0.5909
Polynomial Regression	0.0133	0.1388	0.7728	0.2775	1.4939	1.905	1.0802	0.5717
SGD Regression	-0.0272	0.1238	0.7271	0.2082	1.0700	0.8357	0.9754	0.3443
Ridge Regression	0.0005	0.1102	0.7252	0.2199	1.1338	1.0330	0.9893	0.3937
Lasso Regression	0.0192	0.0952	0.7004	0.2091	0.9643	0.7059	0.9297	0.3162
Elastic Net	0.0192	0.0952	0.7004	0.2091	0.9643	0.7059	0.9297	0.3162
Linear SVR	-0.0183	0.1120	0.7160	0.2113	1.0530	0.8697	0.9628	0.3551
RBF SVR	0.0107	0.0958	**0.6995**	0.2070	**0.9627**	0.7055	0.9292	0.3151
TheilSen Regressor	0.0315	**0.0725**	1.1144	1.0272	13.4861	32.7260	2.2837	2.8758
RANSAC Regressor	0.0263	0.4047	3.2632	2.4255	244.8886	525.5126	9.7407	12.2477
Stacked Model	**0.1260**	0.0887	0.7117	**0.2015**	0.9786	**0.6971**	**0.9401**	**0.3080**

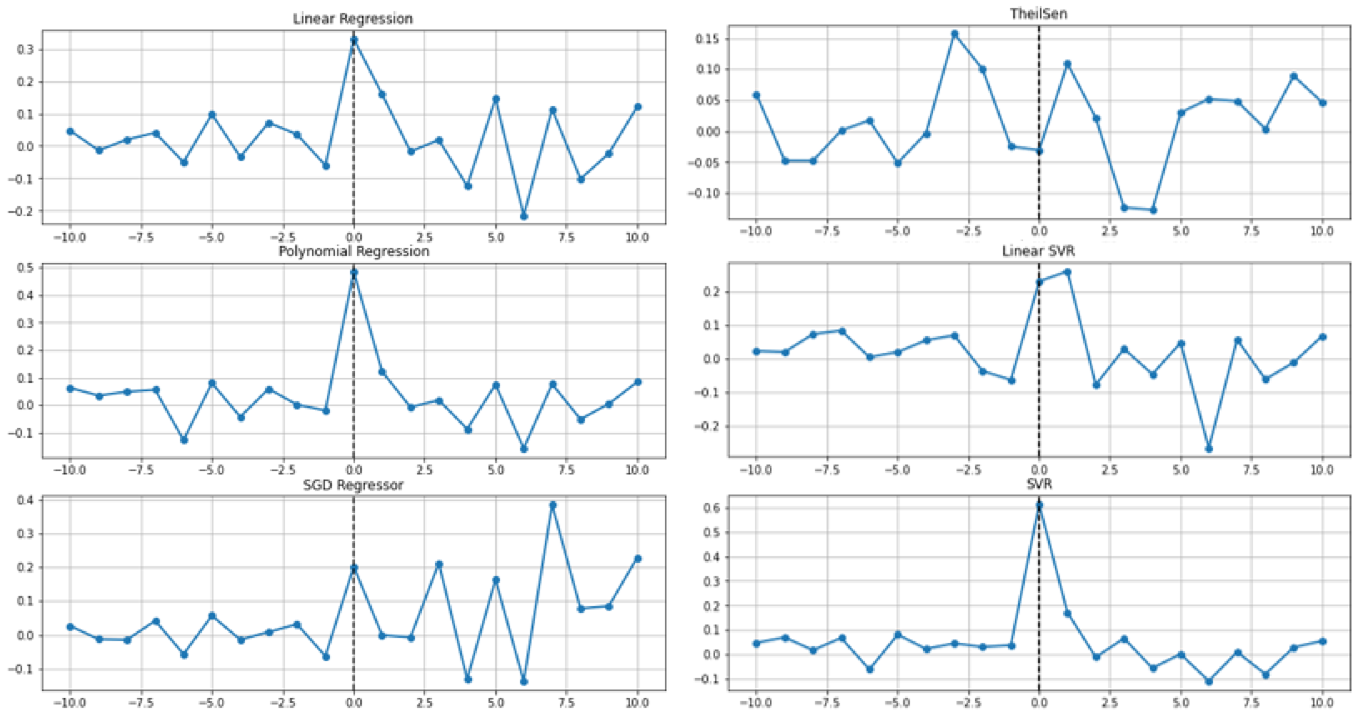

Fig. 4. Cross-correlation plots of models on train set. The plot's x axis is number of days and y axis is correlation value. Polynomial Regression, SGD Regressor and SVM perform best while TheilSen performs worst.

paper, we set split size as 15 and skip first 5 splits to get proper data for training. This method of CV provides an almost unbiased estimate of true error [14].

The models are then evaluated on Mean Error (Bias), Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). An average value and standard deviation of errors are presented, giving an estimate of goodness of the fit of models on folds. All metrics except Bias should be as low as possible. "TABLE II" summarizes the models and their error on CV splits.

The trained models are further analyzed to know what these models have learned and what they are predicting. Most commonly seen learned model by ML Algorithms is persistence model. Given today's price, the model will predict that tomorrow's price will also be same. This is naïve and we want to remove ML models who learned persistence model.

Cross-correlation is a measurement that tracks the changes in two or more time series data in relation to one another to assess how well they match and when the best match takes place.

$$y_i = \sum_j w_j x_{i=j} \tag{26}$$

The spike in cross-correlation plots indicate which day(s) contribute most to the next prediction. A higher correlation at or before the current day (left side of dotted line) would indicate a persistence model - a model that copies the value of current or previous data points. "Figure 4" gives the cross-correlation plots of models described in Section III C.

Fig. 5. Comparison of ML models on test data plotted on candlestick chart.

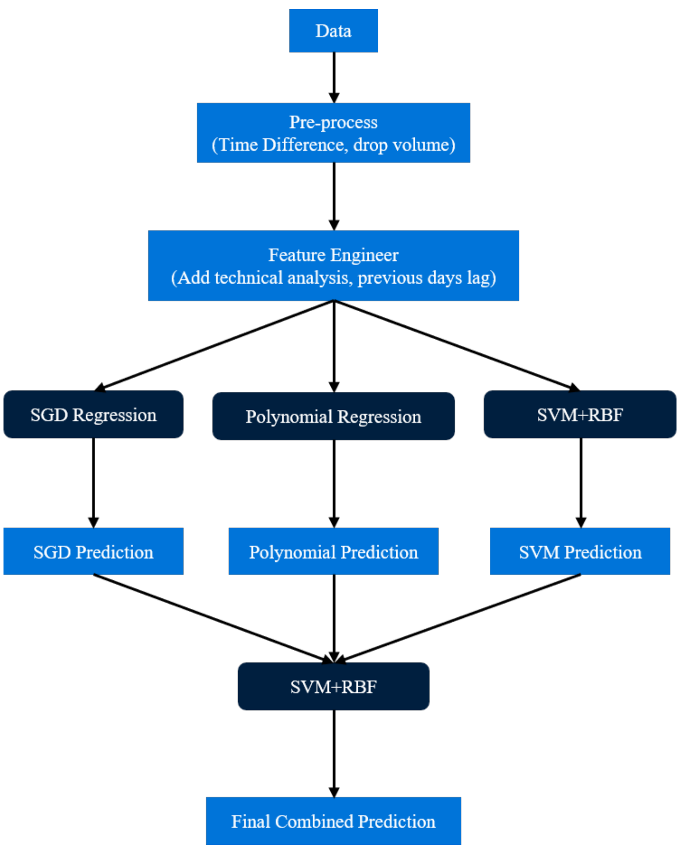

Fig. 6. End-to-end ML pipeline for robustly predicting next data point.

TheilSen and Lasso are predicting the worst and their metrics confirms the same. The best 3 models, Polynomial Regression, SGD Regression and SVM (RBF kernel), are stacked together and trained with SVM regressor to finally combine the output and squeeze the error. "Figure 5" shows the prediction of various ML models experimented with and an ensemble or combination of best 3 models. "Figure 6" shows an end-to-end pipeline to robustly predict time series data

V. Future Work

Tree-based model gets clipped at the highest (or near highest) data point available in the dataset. This is an abnormal behavior in context of stock market. Just because some stock has created a high does not necessarily mean that it is an absolute all time high. This abnormal behavior is removed via time differencing presented in this paper. An interesting extension of this work would be to analyze algorithmic trading rules and the optimal decisions found by tree-based models.

VI. Conclusion

In this work we have compared machine learning models for time series forecasting, discussed about robustly predicting the succeeding data point in time series data and the automation of trading with Machine Learning. We discussed that time differencing is a prominent pre-processing step, often skipped by many papers, and we further analyzed the predictions made by ML algorithms with cross-correlation to verify that they are not replicating the value of current data points (persistence model). At last, we presented a novel architecture to robustly forecast time series data by stacking SGD Regressor, Polynomial Regressor and SVM with SVM model.

References

[1] Box, G.E., Jenkins, G.M., Reinsel, G.C. and Ljung, G.M., 2015. Time series analysis: forecasting and control. John Wiley & Sons.

[2] Hegazy, O., Soliman, O.S. and Salam, M.A., 2014. A machine learning model for stock market prediction. arXiv preprint arXiv:1402.7351.

[3] Shen, S., Jiang, H. and Zhang, T., 2012. Stock market forecasting using machine learning algorithms. Department of Electrical Engineering, Stanford University, Stanford, CA, pp.1-5.

[4] Kim, H.J. and Shin, K.S., 2007. A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets. Applied Soft Computing, 7(2), pp.569-576.

[5] Pai, P.F. and Lin, C.S., 2005. A hybrid ARIMA and support vector machines model in stock price forecasting. Omega, 33(6), pp.497-505.

[6] Ince, H. and Trafalis, T.B., 2004, July. Kernel principal component analysis and support vector machines for stock price prediction. In 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541) (Vol. 3, pp. 2053-2058). IEEE.

[7] Adebiyi, A.A., Ayo, C.K., Adebiyi, M. and Otokiti, S.O., 2012. Stock price prediction using neural network with hybridized market indicators. Journal of Emerging Trends in Computing and Information Sciences, 3(1).
[8] Lee, M.C. and To, C., 2010. Comparison of support vector machine and back propagation neural network in evaluating the enterprise financial distress. arXiv preprint arXiv:1007.5133.
[9] Calabuig, J.M., Falciani, H. and Sánchez-Pérez, E.A., 2020. Dreaming machine learning: Lipschitz extensions for reinforcement learning on financial markets. Neurocomputing, 398, pp.172-184.
[10] J. Welles Wilder, "The Relative Strength Index" in New concepts in technical trading systems, Trend Research, 1978, pp.63-70
[11] Levy, R.A., 1967. Relative strength as a criterion for investment selection. The Journal of finance, 22(4), pp.595-610.
[12] Bollinger, John. "Using bollinger bands." Stocks & Commodities 10, no. 2 (1992): 47-51.
[13] Kite Connect from Zerodha. Link: https://kite.trade/docs/connect/v3/
[14] Varma, S. and Simon, R., 2006. Bias in error estimation when using cross-validation for model selection. BMC bioinformatics, 7(1), pp.1-8.
[15] Vapnik, V.N., 1995. The nature of statistical learning. Theory.
[16] Yang, H., Chan, L. and King, I., 2002, August. Support vector machine regression for volatile stock market prediction. In International Conference on Intelligent Data Engineering and Automated Learning (pp. 391-396). Springer, Berlin, Heidelberg.