Prediction of Stock Market Trends using Machine Learning Techniques

[1] CH. Mani Prasad, [2]O. Mrudhula, [3]Dr. A.M. Sowjanya
[1]M.Tech Scholar, Department of Computer Science and System Engineering,
Andhra University College of Engineering (A), Visakhapatnam, AP, India.
[2] Research Scholar, Department of Computer Science and System Engineering,
Andhra University College of Engineering (A), Visakhapatnam, AP, India
[3]Professor, Department of Computer Science and System Engineering,
Andhra University College of Engineering (A), Visakhapatnam, AP, India

Abstract: *Accurate prediction of Stock Market data is a major research problem in Timeseries community. Anything that is observed sequentially over time is a timeseries. Since Stock Market can make huge impact on an investor's life, so before taking a decision we need to analyze all the factors of stock data. In this paper we proposed an ensemble technique to improve prediction of stock market trends. Exponential smoothing is one of the most popular forecasting methods. Here we use a technique for bootstrap aggregation (bagging) of exponential smoothing methods, which results in significantly improved forecasts. Bagging uses a Box-Cox transformation followed by STL decomposition to separate the timeseries into trend, seasonal part, and remainder.*

Keywords: Timeseries, forecasting, ensemble, bagging, bootstrapping, exponential smoothing, BaggedETS.

I. INTRODUCTION

A TimeSeries is a sequence of observations over a time, which are usually spaced at regular intervals of time. Timeseries forecasting is an important research area In stock market prediction. Timeseries forecasting plays major role to predict future values based on the historical data . Finding accurate prediction is a challenging task in stock market analysis. The timeseries data can include the values collected at the end of every week, month, quarter, or year [1]. Our main idea is to find relationship among any data. we have many types of stock market organizations like BSE, NASDAQ, S&P500, NSE, GSPC, etc.

There are two analytical methods for stock trading Fundamental analysis and Technical analysis. Fundamental analysis focus on the central factors of a company. Such as financial statements, earnings per share, return on assets, price earnings ratio, return on equity, price and book value. On the other hand technical analysis uses the recent historical data of individual stock, including opening, high, low, and closing prices and the volume of the stock, to predict future stock price values[2]. In this paper we perform Technical analysis by using closing values of the historical data to predict the future values.

There are many statistical models available for forecasting stock trends and choosing an appropriate model for particular forecasting technique depends on the format of data. Forecasting techniques like ARIMA, neural network, ets, naïve, snaive,svm, linear regression, random forest, etc…are mostly used for regression. ARIMA(Autoregressive Integrated Moving Avarage) model is the widely used forecasting technique to predict future values based on historical data. we have proposed an baggedets (bagged Exponential smoothing) algorithm to predict future stock values.

ARIMA and Exponential smoothing model is the weight assignment procedure to it's past lag values and error term. In that case Exponential should be considered much better than ARIMA due to its weight assigning method.

Bootstrap aggregating (bagging), as proposed by Breiman (1996), is a popular method in Machine Learning to improve the accuracy of predictors by addressing potential instabilities [3]. If we produce forecasts from each of the additional timeseries, and average the resulting forecasts, we get better forecasts than if we simply forecast the original timeseries directly. This is called "bagging" which stands for "bootstrap aggregating".[4]. ETS stands both for ExponenTial Smoothing and for Error, Trend, and Seasonality, which are the three components that define a model within the ETS family. we propose a bagging methodology for

exponential smoothing method, and evaluate it on the M3 data. As our input data are non-stationary timeseries, both serial dependence and non-stationarity have to be taken into account. We resolve these issues by applying a seasonal-trend decomposition based on loess and a moving block bootstrap on the residuals of the decomposition.

II. RELATED WORK

Autoregressive integrated moving average model(ARIMA)

One of the most important and widely used timeseries models is the autoregressive integrated moving average(ARIMA) model. In an ARIMA model, the future value of a variable is assumed to be a linear function of several past observations and random errors, i.e., A process is said to be an ARIMA of order (p,d, q).[5]

Non-stationary timeseries processes can be transformed, by differencing the series once or more, to make them stationary.The number of times that the integrated process must be differenced to make a timeseries stationary is said to be the order of the integrated process [6].

III. PROPOSED METHODOLOGY

The methodology includes exponential smoothing, and the bootstrapping procedure involving a BoxCox transformation, STLdecomposition.We illustrate the steps using the GSPC dataset, which is a monthly series.We can perform prediction on monthly, daily, or yearly data, so we need to collect data and perform pre-processing.

- **Data Collection:**

We can collect data from any financial websites like Yahoo finance,Google finance etc. Hence GSPC historical stock data from Yahoo finance, which is stock exchange data of US economy has been considered.we use monthly data (1950-2018)total 58 years of data.

- **Pre-processing**

In preprocessing, redundant attributes are removed from data sets. We perform prediction on closing values of monthly data.we have a .doc format of data so we need to convert .doc to .xts format buy using timeseries object then we split data into training set and test sets.

- **BaggedETS**

Bagging is also called Bootstrap Aggregation .The bootstrap is a powerful statistical method for estimating a quantity from a data sample. This is easiest to understand if the quantity is a descriptive statistic such as a mean or a standard deviation. Bootstrap Aggregation (or Bagging for short), is a simple and very powerful ensemble method.An ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model[15].Bootstrap Aggregation is a general procedure that can be used to reduce the variance for those algorithm that have high variance.The Implementation of proposed system can be shown in following figure.1.

- **Exponential smoothing**

The general idea of exponential smoothing is that for forecasting, recent observations are more relevant than older observations, so that they should be weighted more highly. Accordingly, simple exponential smoothing, for example, uses a weighted moving average with weights that decrease exponentially[9]. Starting from this basic idea, exponential smoothing has been expanded for modeling of different components of a series, such as trend, seasonal, and remainder components, where the trend captures the long-term direction of the series, the seasonal part captures repeating components of a series with a known periodicity, and the remainder captures unpredictable components. The trend component is a combination of a level term and a growth term. There is a whole family of ETS models, which can be distinguished by the type of error, trend, and seasonality they use[12].

- **Implementation**

In general, the trend can be non-existent, additive, multiplicative, damped additive, or damped multiplicative. The seasonality can be non- existent, additive, or multiplicative. The error can be additive or multiplicative; however, distinguishing between these two options only has consequences for the prediction intervals, not for the point forecasts [12].

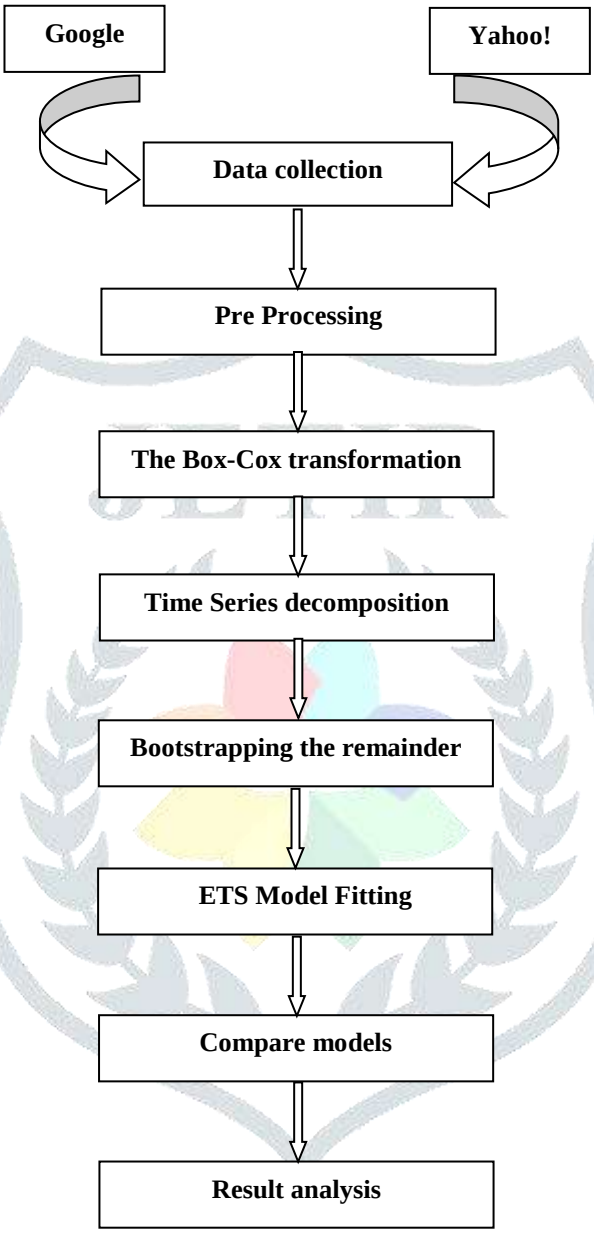


Fig.1. Architecture of Proposed System

- **The Box-Cox transformation**

This is a popular transformation to stabilize the variance of a timeseries, originally proposed by Box and Cox. Depending on the parameter λ, the transformation is essentially the identity (λ = 1), the logarithm (λ = 0), or a transformation somewhere between. A difficulty is the way to choose the parameter λ. In this work, we restrict it to lie in the [0, 1] interval, and then use the method of Guerrero to choose its value in the following way. The series is divided into subseries of length equal to the seasonality or of length two if the series is not seasonal. Then, for each of the subseries, the sample mean m and standard deviation s are calculated, and λ is chosen in a way that the coefficient of variation of s/m(1−λ) across the subseries is minimized.[7][9]

- **Timeseries decomposition**

For non-seasonal timeseries, we use the loess method a smoothing method based on local regressions, to decompose the timeseries into trend and remainder components[9]. For seasonal timeseries, we use STL, as presented by Cleveland to obtain trend, seasonal and remainder component In loess, for each data point a neighborhood is defined[8]. The points in that neighborhood are then weighted (using so-called neighborhood weights) according to their distance from the respective data point. Finally, a polynomial of degree d is fitted to these points. Usually,d = 1 and d = 2 are used, i.e., linear or quadratic curves are fitted. The trend component is equal to the value of the polynomial at each data point. In R, loess smoothing is available by the function loess[11].

In STL, loess is used to divide the timeseries into trend, seasonal, and remainder components. The division is additive, i.e., summing the parts gives the original series again. In detail, the steps performed during STL decomposition are: (i) detrending; (ii) cycle subseries smoothing series are built for each seasonal component, and smoothed separately;(iii)lowpass_ltering of smoothed cycle-subseries | the sub-series are put together again, and smoothed; (iv) de trending of the seasonal series; (v) deseasonalizing the original series, using the seasonal component calculated in the previous steps; and (vi) smoothing the deseasonalized series to get the trend component. In R, the STL algorithm is available through the stlfunction[12] .

Another possibility for decomposition is, to use ETS modelling directly, as proposed by Cordeiro and Neves (2009). However, an ETS model has its components defined in terms of the noise terms, and they dynamically evolve with the noise. So, “simulating” an ETS process by decoupling the level, trend and seasonal components from the noise and treating them as if they are independent series may not work well. This is in contrast to an STL decomposition in which the trend and seasonal components are smooth and the way they change over time does not depend directly on the noise component. Therefore we can independently simulate the noise term in an STL decomposition using bootstrapping procedures[10].

- **Bootstrapping the remainder**

As timeseries data are typically auto correlated, adapted versions of the bootstrap exist (see Lahiri, 2003; Gon¸calves and Politis, 2011). A prerequisite is stationarity of the series, which we achieve by bootstrapping the remainder of the STL (or loess) decomposition[13].

In the MBB as originally proposed by Ku¨nsch (1989), data blocks of equal size are drawn from the series, until the desired series length is achieved. For a series of length n, with a block size of l, n l + 1 (overlapping) possible blocks exist.We use block size of l =22 for monthly data, i.e., at least two years, to ensure any remaining seasonality is captured. As the shortest series has n = 12 observations in total for the yearly data, care has to be taken that every value from the original series can possibly be placed anywhere in the bootstrapped series. To achieve this, we draw n/l + 2 blocks from the remainder series. Then, we discard from the beginning of the bootstrapped series a random amount between zero and l -1 values. Finally, to obtain a series with the same length as the original series, we discard the amount of values necessary to obtain the required length. This processing ensures that the bootstrapped series does not necessarily begin or end on a block boundary[14].

After bootstrapping the remainder, the trend and seasonality are combined with the bootstrapped remainder, and the Box-Cox transformation is inverted, to get the final boot- strapped sample. After generating the bootstrapped time series, to every series the ETS model fitting procedure is applied. By applying the whole ETS fitting and model selection procedure to every bootstrapped time series independently, we address data uncertainty, parameter uncertainty, and model selection uncertainty[7][9].For each horizon, the final resulting forecast is calculated from the forecasts from the single models. We performed preliminary experiments using the mean, trimmed mean, and median.

- **The Bagged ETS Method**

we develop this project by using R language, and development tool is R Studio.R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.R

provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering etc) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity[10].

RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

We could simply average the simulated future sample paths computed earlier,our interest is only in improving point forecast accuracy.We will use ets() to forecast each of these series.The average of these forecasts gives the bagged forecasts of the original data[4][10].The above whole procedure can be done by using the function BaggedETS( ).The length of the blocks used for obtaining bootstrapped residuals is set to 22 for monthly data set[4].

IV. RESULTS

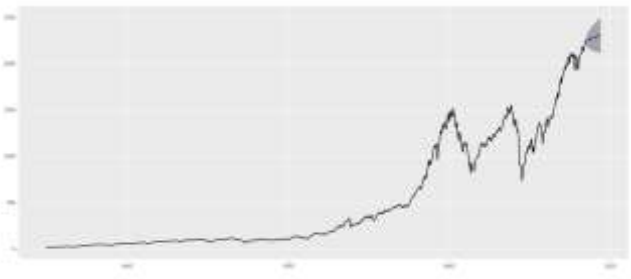
In this section, we describe the forecasting methods, error measures, and statistical tests that were used in the experiments, and the results obtained on the GSPC dataset.

1. **Compared Methods**

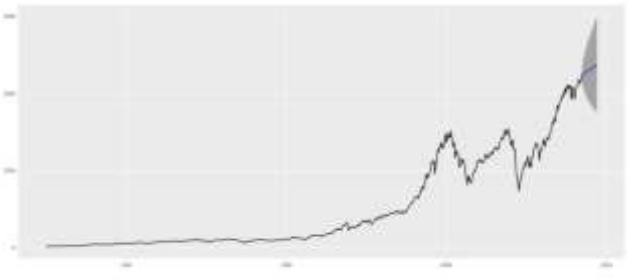
We call the decomposition approach of Box-Cox transformation and STL or loess proposed in this paper *Box-Cox and loess-based decomposition (BLD)*.We compare our proposed method to the original ETS method, as well as ARIMA model,Naïve forecast Neural-Network.

		ME	RMSE	MAE
ARIMA	Training set	0.495	30.341	16.308
	Test set	348.893	381.0896	348.893
FITETS	Training set	1.372	30.349	15.999
	Test set	314.463	343.294	314.463
NEURAL NETWORK	Training set	0.595	30.438	15.977
	Test set	525.603	594.884	525.603
BAGGED ETS	Training set	0.398	29.074	15.178
	Test set	221.647	257.919	229.732

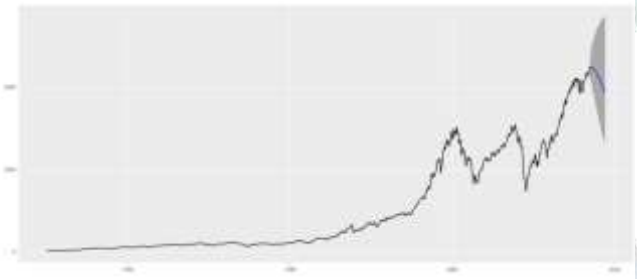
Fig(1).Final Result of the Algorithms



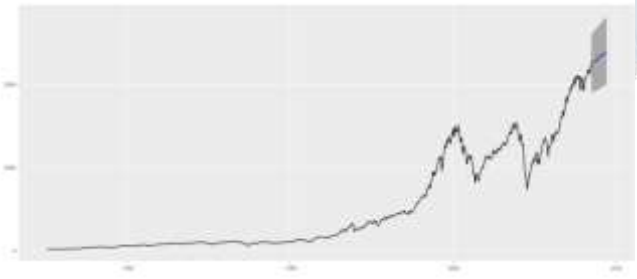
Fig(2).ARIMA model ggplots



Fig(3).ETS model ggplots



Fig(4).Neural network model



Fig(5).Bagged ETS model

V. CONCLUSION

We have compare the ARIMA,NEURAL NETWORK, ETS, BAGGED ETS forecasting algorithms the BaggedEts algorithm has given better results by comparing ME, RMSE , MAE values.we have proposed an ensemble method to predict stock market indexes.The results indicate that Ensemble method performs better than ARIMA Model. we have proposed a method to improve the accuracy of the prediction model for better forecasting of the stock values.

REFERENCES

[1]. Mahantesh C. Angadi, AmoghP.Kulkarni "TimeSeries Data Analysis for Stock Market Prediction using Data Mining Techniques with R" IJARCS, July-August, 2015, 104-108

[2]. SornponWichaidit, SurinKittitornkun, "Predicting SET50 Stock Prices Using CARIMA(Cross Correlation ARIMA)" 978-1-4673-7825-3/15/$31.00 @2015 IEEE

[3]. Christoph Bergmeira, ∗, Rob J Hyndmanb , Jos´e M Ben´ıtezc" Bagging Exponential Smoothing Methods using STL Decomposition and Box-Cox Transformation" International Journal of Forecasting, Wednesday 22nd July, 2015

[4]. https://otexts.org/fpp2/bootstrap.html

[5]. Fengxia Zheng, ShoumingZhong," Timeseries forecasting using an ensemble model incorporating ARIMA and ANN based on combined objectives" 978-1-4577-0536-6/11/$2 ©2011 IEEE

[6]. Rob J Hyndman, Anne B Koehler , Ralph D Snyder, Simone Grose (2002) International Journal of Forecasting 18(3), 439-454

[7]. Box, G.E.P., Cox, D.R., 1964. An analysis of transformations. Journal of the Royal Statistical Society,Series B 26, 211–252IEEE

[8]. Cordeiro, C., Neves, M., 2009. Forecasting timeseries with BOOT.EXPOS procedure. REVSTAT - Statis- tical Journal 7, 135–149.IEEE

[9]. Hyndman, R., Athanasopoulos, G., 2013. Forecasting: principles and practice. URL: http://otexts.com/fpp/.

[10]. Hyndman, R., Khandakar, Y., 2008. Automatic timeseries forecasting: The forecast package for R. Journal of Statistical Software 27, 1–22.

[11]. Hyndman, R., Koehler, A., 2006. Another look at measures of forecast accuracy. International Journal of Forecasting 22, 679–688.

[12]. Hyndman, R., Koehler, A., Snyder, R., Grose, S., 2002. A state space framework for automatic forecasting using exponential smoothing methods.International Journal of Forecasting 18, 439–454.

[13]. Hyndman, R.J., 2013. Mcomp: Data from the M-competitions.URL: http://robjhyndman.com/software/mcomp/.

[14]. Hyndman, R.J., Koehler, A.B., Ord, J.K., Snyder, R.D., 2008. Forecasting with Exponential Smooth- ing: The State Space Approach.Springer Series in Statistics, Springer. URL: http://www. exponentialsmoothing.net.

[15]. Kourentzes, N., Barrow, D., Crone, S., 2014a. Neural network ensemble operators for timeseriesforecasting.Expert Systems with Applications 41, 4235–4244.