Prediction of Stock Market using C-means Clustering and Particle Filter

Ahmed Haj Darwish
Associate Professor
Dept. Artificial Intelligence and Natural Languages
Faculty of Informatics Engineering
University of Aleppo, Syria

Aliaa Hilal
Master Student
Dept. Artificial Intelligence and Natural Languages
Faculty of Informatics Engineering
University of Aleppo, Syria

ABSTRACT

In this article, Particle Filter and C-means are used to predict a value of a point in a time series. Similar data in a time-series are grouped using C-means algorithm. Afterward, a number of particle filters are used as sub-predictors. These sub-predictors start from different points, which are the centers of clusters resulted from clustering algorithm. Outputs from all filters were used to obtain Final prediction result. A weighted average method is used to aggregate the outputs of the filters. Particle filters are used in here to model non-Gaussian time series. Benchmark datasets were used to evaluate the proposed algorithm. To measure its prediction performance, the results derived from the proposed model were compared with those of other algorithms. The comparison proved the effectiveness and accuracy of the proposed method.

General Terms

Machine Learning, Time Series.

Keywords

Prediction, Time Series, C-means, Particle filter, Stock price, Importance Resampling.

1. INTRODUCTION

Over the past two decades, many important changes have taken place in financial markets. The development of powerful communication and trading facilities has enlarged the scope of selection for investors. Investors and shareholders always seek to predict stock prices precisely. Therefore, accurate modelling and prediction of values in such a nonlinear system, ex. stock market, is a difficult task. This prediction helps investors to determine the appropriate timing for sale and purchase of shares. As the Prediction process is based on the principle that history repeats itself, the future stock prices can be determined using those of previous observations [1].

The behavior of this kind of systems is usually modeled by time series, where time series is ordered observations describing changes in a process over time [1]. This model can be used to predict a value of a point in the time series for a given time, this process is also known as time series prediction. Time series is usually non-linear or chaotic [2]. Estimation of the future values, which is known as forecasting, is also classified as prediction. In prediction of a time series, successor values are used to estimate values of predecessor values.

The accuracy of time series prediction is very important. Therefore, to trust and to adopt this prediction, the estimation of values should be accurate with low error [3]. One of the most important benefits of time series prediction is to obtain knowledge about changing a certain process before its occurring, this knowledge may help in take constructive and meaningful decisions.

Stock price prediction has spurred the interest of researchers over the years to develop better predictive models. The most used techniques fall into two broad categories, namely, statistical and Machine Learning techniques [1]. Statistical techniques include, among others, exponential smoothing, autoregressive integrated moving average (ARIMA) [4]. The ARIMA model, also known as the Box-Jenkins model, is commonly used in analysis and forecasting. In [4], the ARIMA model was used as stock price predictive model. New York Stock Exchange (NYSE) and Nigeria Stock Exchange (NSE) data [4] were used to test the ARIMA model. Results obtained revealed that the ARIMA model has a strong potential for short-term prediction. The forecasting performance of ARIMA and artificial neural networks model [5] has been examined with New York Stock Exchange data. The empirical results obtained prove the superiority of neural networks model over ARIMA model. However, ARIMA gives good results for linear time series. However, time series data in real-world commonly have nonlinear features under a new economic era. Consequently, ARIMA may be unsuitable for most nonlinear real-world problems.

A number of machine-learning techniques have been used to avoid restrictions of statistical techniques in stock price prediction. Artificial Neural Networks (ANN) [6] were utilized feed-forward architecture to predict share market price. The ANN prediction model was trained using a dataset of Microsoft Corporation for year 2011 [6], which contains open, high, low, adjacent close and volume as input parameters and close price as output parameter (predict value). Hidden Markov Models (HMM)-based solution has been used to predict the stock market fluctuation [7], the implemented algorithm was tested on three different stock indices ICICI, SBI, IDBI. Mean Absolute Percentage Error was used as a metric to evaluate the performance of the algorithm. Lin, Guo, and Hu [8] proposed a SVM-based system for stock market prediction. This system selects a good feature subset, evaluates stock indicators. This approach was tested using Taiwan stock market datasets. This system performed better than the conventional system.

Gupta, Aditya, and Dhingra [9] proposed a stock market prediction technique based on Hidden Markov Models (HMM). In this approach, fractional changes in stock values and the intra-day high and low values of the stock was used to train the continuous HMM. After that, this HMM was used to make a decision over all the possible stock values for the next day. Mean Absolute Percentage Error (MAPE) was used to compare the performance of this approach and other existing methods on several stocks. Gupta and Sharma [10] proposed hybrid combinatorial method of clustering and classification for prediction of Shanghai stock market values. The method first clustered stock market values using K-means clustering algorithm and these clustered values are classified using horizontal partition decision tree. Nguyen [11] used Hidden Markov Models (HMM) with both single and multiple observations to forecast economic regimes and stock prices. HMMs in this article test to predict S&P500 closing price for one year from July 2012 to July 2013. The experimental result of HMM using multiple observations has a relative error smaller than the error of using one observation data. Wei [12] proposed a hybrid time series adaptive network-based fuzzy inference system (Adaptive Neuro-Fuzzy Inference System ANFIS) model to forecast stock prices. ANFIS were presented [13] with a hybrid training algorithm integrating the Bees Algorithm (BA) and Least Square Estimation (LSE). This model with three inputs was used to predict the value of Mackey-Glass time series as output. Xu and Zhang [14] suggested the use of Kalman filter to predict the price of the shares of Changbaishan, where Kalman filter has dynamic tracking features and well during the real time.

In this article, an efficient approach is devised for stock market prediction by employing C-means clustering and Particle filter. C-means algorithm is used at the beginning of the proposed algorithm to improve the accuracy of stock market prediction, where stock market data is characterized with large size, multi-dimensional and uncertainty [15]. C-means algorithm divides the data into set of clusters, so that it simplifies the prediction of stock market and determines the appropriate input for the Particle filter. Where data points from all clusters are used as input for Particle filter.

The rest of the paper is organized as follows. Section 2 defines Time Series. Next, in Section 3 C-means clustering algorithm is described, Particle filter is explained in section 4. Section 5 illustrates the proposed method. Finally, experimental results of the proposed algorithm are provided in Section 6. These results are discussed and analyzed in Section 7 and conclusion is given in Section 8.

2. TIME SERIES

Time series is a record of the observed values of a process taken sequentially over time [1]. It is mathematically defined as a collection of random variables {Y (t); t ∈T} where T is an index set (represents time) for which all the random variables Y (t), t∈ T, are defined on the same sample space. Time series data have a natural temporal ordering. This makes time series analysis distinct from other common data analysis problems [15]. A time series containing points of a single attribute is termed as univariate time series. However, if the points of more than one attribute are considered, it is termed as multivariate time series [16]. In continuous time series, observations are measured at every instance of time like temperature readings and flow of a river, whereas a discrete time series contains observations measured at discrete points of time like population of a particular city and production of a company [16]. Usually, in a discrete time series the consecutive observations are recorded at equally spaced time intervals such as hourly, daily, weekly, monthly or yearly time separations [17]. Fig.1 shows international airline passenger time series as an example.

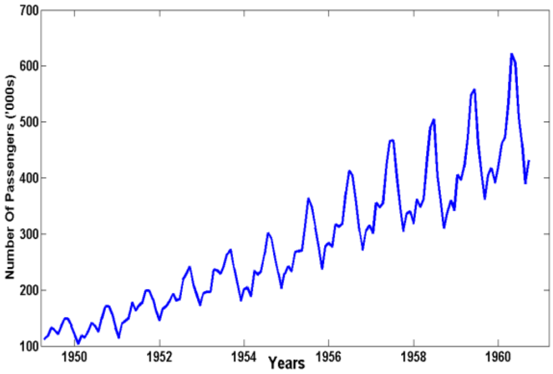

Fig.1: Monthly international airline passenger series (Jan. 1949-Dec. 1960)

3. C-MEANS ALGORITHM

C-means algorithm was introduced by Dunn, 1973 and later extended by Bezdek, 1981 [18]. It has been widely used in cluster analysis, pattern recognition and image processing. C-means algorithm groups data by assigning a membership value linking each data point to a cluster center. Unlike K-means where data points exclusively belong to one cluster, Data points in C-means algorithm have membership values to centers of clusters and as such can belong to more than one cluster at a time. Therefore, C-means clustering produces better results for overlapped dataset compared with K-means clustering [19].

C-means algorithm is one of the most popular fuzzy clustering methods based on minimization of a least-squared error function determined in equation (1). It is based on minimizing the criterion with respect to the membership value μ_{ij} and the distance d_{ij} [20].

$$(X;U,C) = \sum_{i=1}^{N}\sum_{j=1}^{c}(\mu_{ij})^{m}\, d_{ij}^{2} \qquad (1)$$

Subject to:
$$\begin{cases} \sum_{j=1}^{c} \mu_{ij} = 1;\ i = 1,2,\ldots.,\ N \\ 1 \geq \mu_{ij} \geq 0;\ i = 1,2,\ldots.,\ N,\ j = 1,2,\ldots,\ c \end{cases} \qquad (2)$$

where N is the number of objects and c is the number of clusters, $\mu_{ij} \in [0,1]$ is the membership degree of data point $x_i$to the j -th cluster, m>1 is the fuzzy factor (weighting exponent which determines the fuzziness of the resulting clusters).

For a given dataset $X = \{x_1, x_2, \ldots., x_N\} \subseteq R^{N*q}$, where N is the number of samples, q is the dimension of the sample, $U = \{\mu_{ij};\ i = 1,2,\ldots,\ N,\ j = 1,2,\ldots,\ c\}$ which is the membership matrix, has to satisfy the constraints in (2), c_j is the center of cluster j. The following steps can summarize C-means algorithm [21]:

Step1: Initialize membership matrix $U = \lfloor \mu_{ij} \rfloor$ with the initial value $U^{(0)}$ and set k=1.

Step2: At k-step, calculate the centers of clusters $c_j^k, j = 1,2,....c$ with $U^{(k-1)}$,

$$c_j^k = \frac{\sum_{i=1}^{N} \mu_{ij}^{(k-1)^m} x_i}{\sum_{i=1}^{N} \mu_{ij}^{(k-1)^m}}, 1 \le j \le c \qquad (3)$$

Step3: Update membership matrix $U^{(k-1)}$ to $U^{(k)}$

$$\mu_{ij}^k = \frac{1}{\sum_{l=1}^{c} \left(\frac{d_{ij}}{d_{il}}\right)^{2/(m-1)}}, 1 \le i \le N, 1 \le j \le c \qquad (4)$$

Step4: if $\left\| U^{(k)} - U^{(k-1)} \right\| < \varepsilon$ then stop, or set k=k+1 and go to step2.

The basic idea of C-means algorithm is that use iterative method for solving equations (3) and (4), until a termination condition is met. Here, ε is the threshold of the termination condition.

4. PARTICLE FILTER

Kalman filter (KF), which is a state estimation technique, is commonly used to approximate linear systems with Gaussian noise. The effectiveness of KF has been demonstrated in many works [14]. Kalman filter was modified to fit nonlinear systems with Gaussian noise, e.g. extended Kalman filter (EKF) and unscented Kalman filter (UKF) [22], [23]. However, the performances of these modified KFs depend on the considered system. Poor state estimation is a result of high nonlinearity. Particle filter (PF) is more suitable for nonlinear system or with non-Gaussian noise because PF neither requires the system to be linear nor assumes that the noise is Gaussian [24].

Particle filter is a Monte Carlo technique for the solution of the state estimation problem [25]. PF is a useful tool for a variety of problems (Tracking, Image processing, smart environments). The key idea is to represent the required posterior density function by a set of random samples (particles) and its associated weights. These samples and its weights are used to compute the estimates. As the number of samples becomes very large, this Monte Carlo representation becomes an equivalent representation of the posterior probability function, and the solution approaches the optimal Bayesian estimate [24].

Sequential Importance Sampling (SIS) algorithm for Particle filter is presented in Fig.2, which includes a resampling step at each instant. The importance density in SIS algorithm represents the posterior density in the present case, where it is a density proposed to represent (PDF) that cannot be exactly computed. After that, particles are drawn from the importance density instead of the actual density [26].

Particle filter computes the state estimate recursively and involves three steps as obtained in Fig.2:

Prediction: the filter uses a previous state to predict the current state based on a given system model.

Correction or Update: the filter uses the measurement of the current sensor to correct the state estimate.

Resampling: the filter also redistributes periodically, or resamples, the particles in the state space to match the posterior distribution of the estimated state. In this step, only the particles with large weights are selected to act as the parent particles while the small-weight particles are eliminated.

A discrete state hypothesis is represented by a particle. However, all particles are used to determine the final state estimate. The estimated state consists of all the state variables [27].

Pseudo code of a Particle filter is as the following [26]:

Step1: Initialize the particles and weights

$$x_0^i \sim P_{x_0}, w_0^i = 1/N; i = 1,2,....N \qquad (5)$$

Step2: Predict the next particles

$$x_t^i = f(x_{t-1}^i); i = 1,2,...., N \qquad (6)$$

Step3: Update the weights by the likelihood (7) and then normalize the weights (8)

$$w_t^i = w_{t-1}^i P(y_t / x_t^i) = w_{t-1}^i P(y_t - h(x_t^i)); i = 1,2,...., N \qquad (7)$$

$$w_t^i = \frac{w_t^i}{\sum_{i=1}^{N} w_t^i} \qquad (8)$$

then an approximation of state is given by equation (9)

$$\hat{x}_t = \sum_{i=1}^{N} w_t^i x_t^i \qquad (9)$$

Step4: If $N_{eff} < N_{th}$ then take samples with replacement from the $\{x_t^i\}_{i=1}^{N}$ set, where the probability to take sample i is w_t^i and set the weights $w_t^i = 1/N; i = 1,2,....N$

$$N_{eff} = \frac{1}{\sum_{i=1}^{N} (w_k^i)^2} \qquad (10)$$

Step5: Set t=t+1 and iterate to step 2.

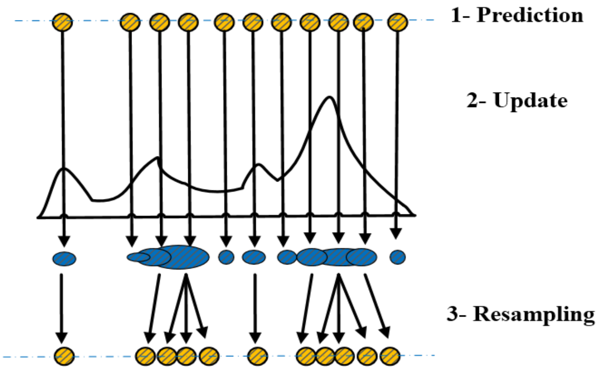

Fig.2: Importance Resampling

5. THE PROPOSED METHOD

Similar to method presented in [23], the proposed method consists of two main stages. At the beginning, the data of a time series which represent stock market prices during a certain period is clustered using C-means algorithm. The reason to use this type of fuzzy clustering that many of time series are characterized by uncertainty. The centers of clusters resulted from C-means are the starting points of particle filters. Despite, hard clustering algorithms, which give only one starting point (a point of time-series data belong to only one cluster). Afterwards, Particle filters are implemented to predict stock market price for next day. Particle filters are used in here as sub-predictors instead of extended Kalman filters which are implemented in [23].

The basic outer structure of the proposed algorithm is illustrated by a flowchart described in Fig.3.

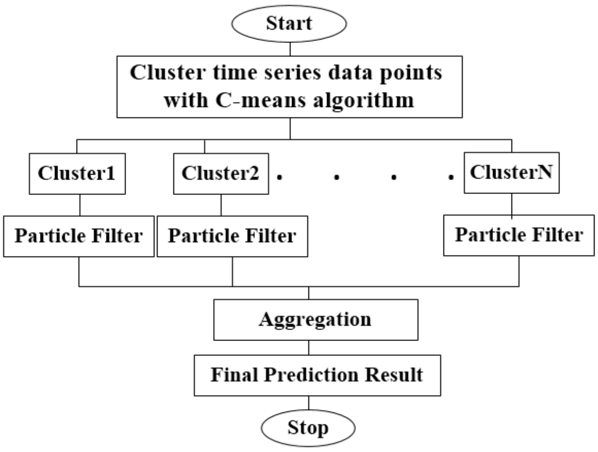

Fig.3: The Proposed Algorithm

Steps of the proposed algorithm:

1. Clustering data points of time series:
   C-means algorithm is used to group similar points of time series data.
2. Execute Particle filter:
   in this step, more than one Particle filter work as sub predictor according to the number of clusters starting from several points. These points are the centers of clusters.
3. Obtain the final prediction result:
   the results from all sub predictor in previous step are used to obtain the final value of prediction.

The number of clusters for C-means algorithm is determined by fuzzy validity indices, such as (Fukuyama-Sugeno(FS) [28], Xie_Beni(XB) [28], Kwon(K) [28], Overlapping and Separation(OS) [28]) which described in equations (11)(12)(13)(14). Optimal number of clusters is calculated in accordance with the criterion of the former indices based on the algorithm described in the following steps:

For each value c=2, 3, …, $\sqrt{N}$ repeat step1 and step2.

Step1: run C-means algorithm with number of clusters c.

Step2: calculate a validity index value for number of clusters c.

Step3: determine the optimal number of clusters, which give minimum value for validity index.

$$XB = \frac{\sum_{i=1}^{N}\sum_{j=1}^{c}\mu_{ij}^{m}\left\|x_i - C_j\right\|^2}{N * D_{\min}^2} \qquad (11)$$

$$FS = \sum_{i=1}^{N}\sum_{j=1}^{c}\mu_{ij}^{m}\left(\left\|x_i - C_j\right\|^2 - \left\|C_j - \bar{C}\right\|^2\right) \qquad (12)$$

$$K = \frac{\sum_{i=1}^{N}\sum_{j=1}^{c}\mu_{ij}^{m}\left\|x_i - C_j\right\|^2 + \frac{1}{c}\sum_{i=1}^{c}\left\|C_i - \bar{C}\right\|^2}{D_{\min}^2} \qquad (13)$$

$$OS = \frac{1}{N}\sum_{k=1}^{N} OS(u_k(x_k), c);$$
$$OS(u_k(x_k), c) = \frac{O(u_k(x_k), c)}{S(u_k(x_k), c)} \qquad (14)$$

Particle filter is chosen among other Bayesian filters because the stock price data are non-linear with non-Gaussian noise. Each filter predicts the value of a next point in a time series based on the data points of each cluster that represents the values of previous time-series data points. The value of a next point in a time series that is determined using cluster data (similar values) is more accurate than this determined using the entire data.

Time series used in this article is a multivariate time series so that three multivariate normality tests, including Mardia‘s [29], Henze-Zirkler‘s [30] and Royston‘s [31] multivariate normality tests are used to test normality of time series data.

The outputs of sub-predictors in the previous step are aggregated to obtain the final result instead of using single sub-predictor output. This aggregation is used to reduce the percentage of error in predicting the value of a future point of time series. Aggregation is implemented via weighted average defuzzification method for sub-predictors outputs (particle filters results) as shown in equation (15).

$$FinalOutptu = \frac{\sum_{i=1}^{N}\mu_i Z_i}{\sum_{i=1}^{N}\mu_i} \qquad (15)$$

where N is the number of sub-predictors equal to the number of clusters in C-means algorithm, z_i is the output of sub-predictor for cluster i, μ_{ij}: is the membership degree of current data point to the i-th cluster, and membership degrees satisfied the equation 2.

6. EXPERIMENTAL RESULTS

The proposed algorithm implemented using Matlab R2014 on computer with the following specifications (CPU: Intel Core i3, System: Windows 7 Ultimate 64- bit, RAM: 2GB) to obtain the future predictions for prices of some financial indices [32]. C-means algorithm implemented with the following parameters (m=2, ε=1e-3, number of iteration=1000, distance function=Euclidian distance) and number of clusters determined based on the values of validity indices, shown in Table (1). After repetition of the algorithm implementation for calculating the number of clusters 10 times for each of the validity index. The optimal number of clusters is determined by the most frequent value between the validity indices. Whereas the results of multivariate normality tests are summarized in Table (2), which indicate that time series data is non-normally distributed. Particle filter parameters are (number of particles=1000, state bounds=bounds of each cluster, resampling method=multinomial).

The first index is S&P 500 index during the period (7/30/2012- 7/31/2013) with four features (close, open, high, low) to predict close price of the share for next day. The prediction accuracy was 0.0036 using Mean Absolute Percentage Error (MAPE) which is described in equation (16) as a performance measure. Fig.4 shows the actual values of index (S&P500) and the prediction results of the proposed algorithm, where the error of prediction is shown in Fig.5.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|M_i - P_i\right|}{M_i} \qquad (16)$$

where M is the actual value of stock price, P is the predicted value for stock price and N is number of predicted points.

A dataset for Microsoft Corporation (MSFT) (from 1/1/2011 to 31/12/2011) was used for experiments. There are six parameters in the dataset open, high, low, volume, adjacent close and close to predict the close value of index for next day. The prediction accuracy is 0.0024 using Mean Square Error (MSE) as given in equation (17). The proposed algorithm was also applied to a number of stock price datasets for companies Apple, IBM, Dell, Tata Steel (from 13/9/2004 to 21/1/2005) with four features (open, high, low, and close) to predict the close value of share for next day. The prediction accuracy is (Tata Steel=1.105, IBM=0.437, Dell=0.538, Apple=1.203) using Mean Absolute Percentage Error (MAPE) as given in equation (16). Fig.8 shows the actual values of IBM index and prediction results of the proposed algorithm, prediction error for this index is shown in Fig.9. Fig.10 shows the actual values of DELL index and prediction results of the proposed algorithm, prediction error for this index is shown in Fig.11. Fig.12 shows the actual values of AAPL index and prediction results of the proposed algorithm, prediction error for this index is shown in Fig.13. Fig.14 shows the actual values of Tata Steel index and prediction results of the proposed algorithm, prediction error for this index is shown in Fig.15.

Table 1: Number of clusters for benchmark datasets according to validity indices XB, K, OS, FS

Benchmark datasets	**Optimal value for XB**	**Optimal value for K**	**Optimal value for OS**	**Optimal value for FS**	**Number of clusters**
S&P 500 [32]	2	2	2	2	2
MSFT [32]	5	3	5	5	5
Tata Steel [32]	6	2	5	2	2
Apple [32]	5	2	2	2	2
IBM [32]	7	3	2	2	2
Dell [32]	10	3	2	3	3

Table 2: Normality tests for benchmark datasets

Benchmark datasets	**Mardia Test**	**Henze-Zikler Test**	**Royston Test**
S&P 500 [32]	X	X	X
MSFT [32]	X	X	X
Tata Steel [32]	X	X	X
Apple [32]	X	X	X
IBM [32]	X	X	X
Dell [32]	X	X	X

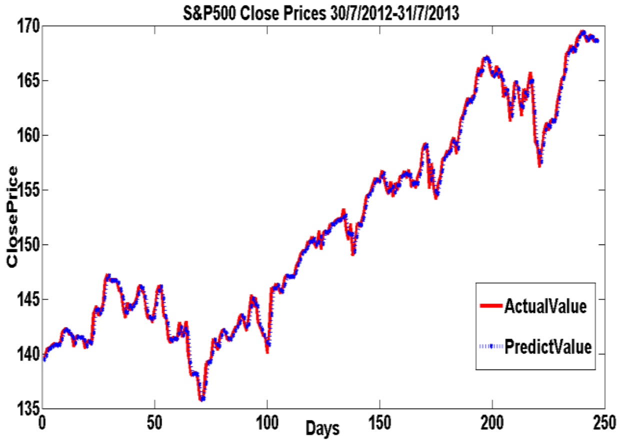

Fig.4: Prediction with SP&500 Results

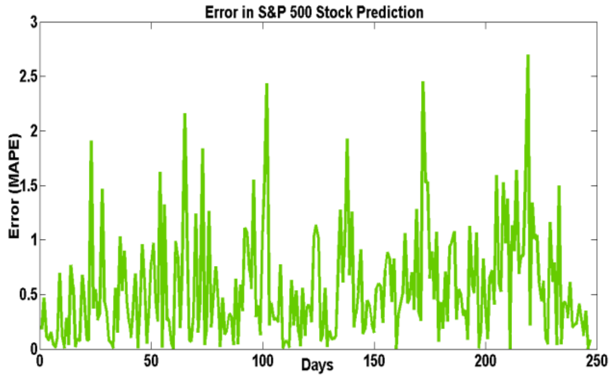

Fig.5: Error in S&P500 Prediction

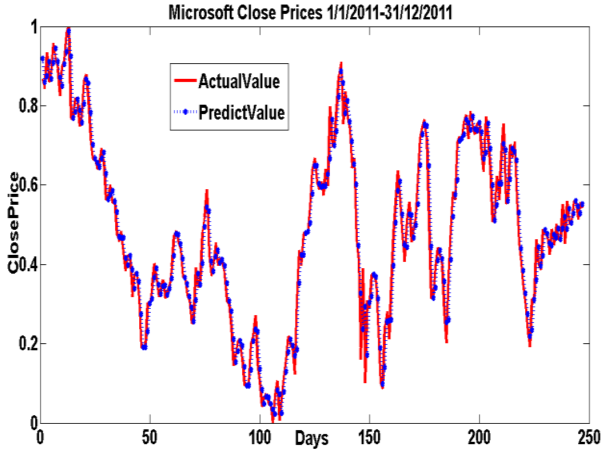

Fig.6: Prediction with MSFT Results

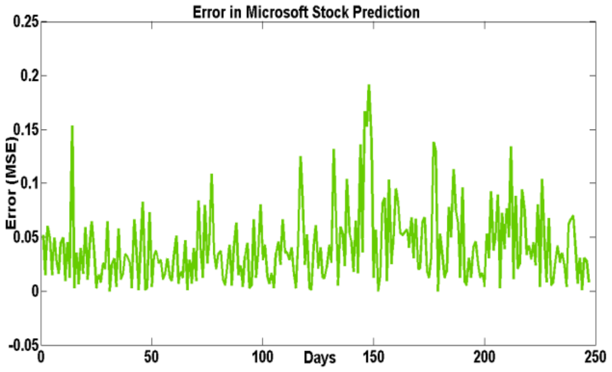

Fig.7: Error in MSFT Prediction

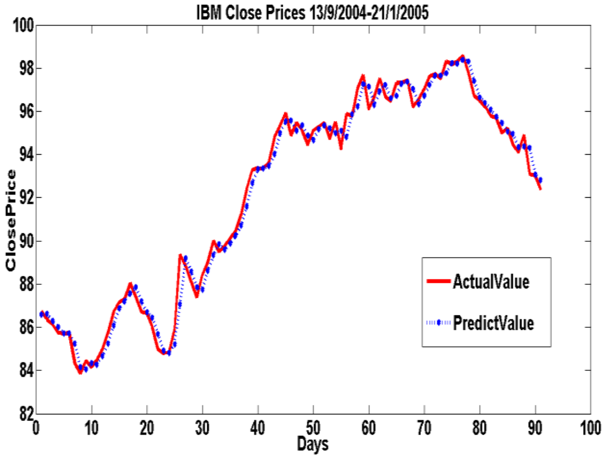

Fig.8: Prediction with IBM Results

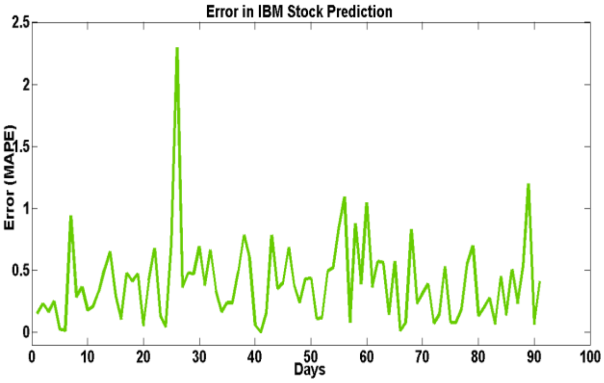

Fig.9: Error in IBM Prediction

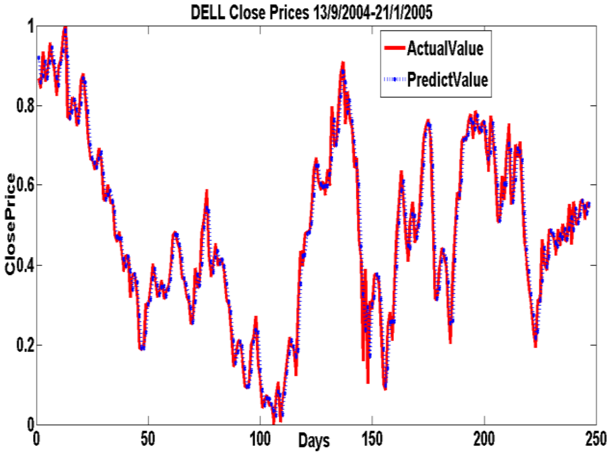

Fig.10: Prediction with DELL Results

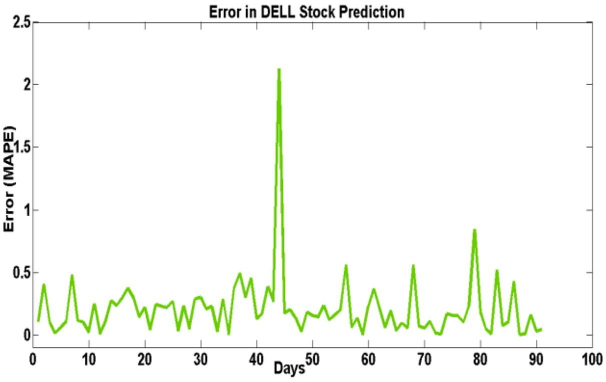

Fig.11: Error in DELL Prediction

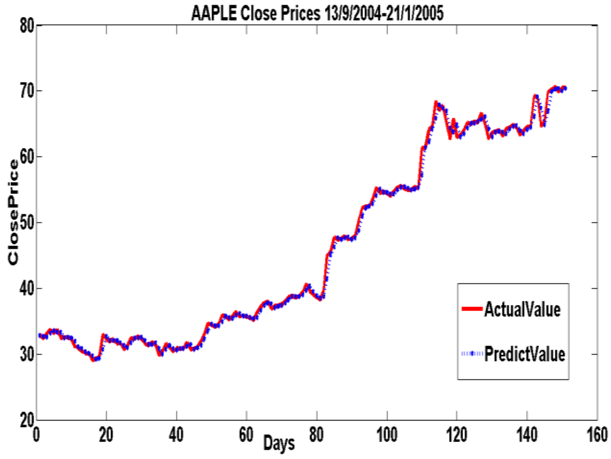

Fig.12: Prediction with AAPL Results

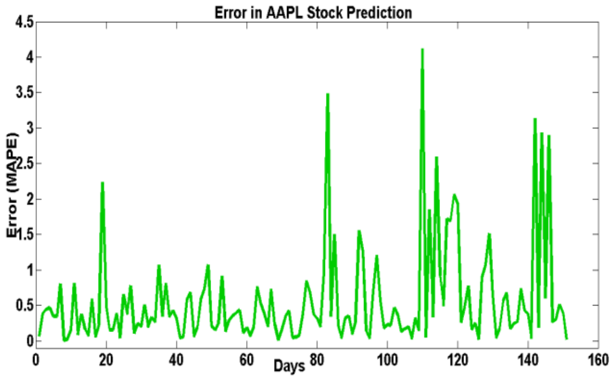

Fig.13: Error in AAPL Prediction

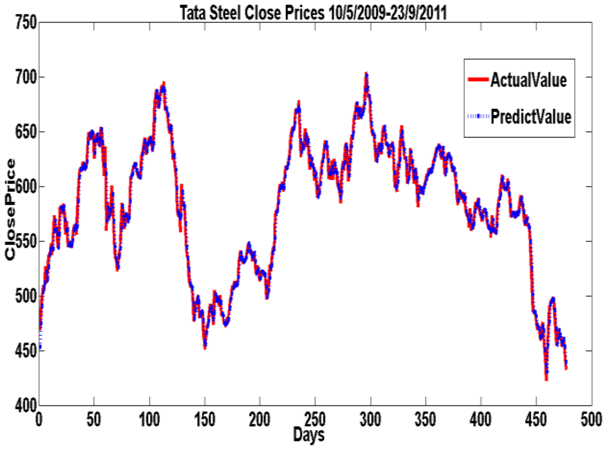

Fig.14: Prediction with TATA STEEL Results

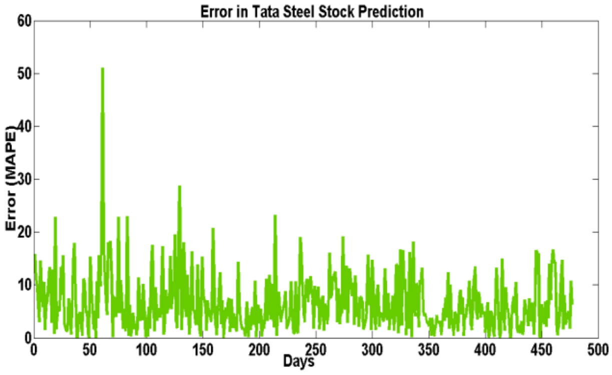

Fig.15: Error in Tata Steel Prediction

7. DISCUSSION AND ANALYSIS RESULTS

Table 3 compares the prediction results of PF & C-means (proposed) with ANN model [6] and EKF & C-means model (proposed [23]) to predict close price for Microsoft Corporation index. Also PF & C-means (proposed) results compared with those of Hidden Markov Model (HMM) [11] and EKF & C-means model (proposed [23]) to predict close price for S&P500 index, the results is shown in Table 4.

Table 3: Comparison of MSFT index prediction performance (MSE) for the proposed algorithm and other algorithms

Algorithm	**MSE**
ANN with trailm [6]	0.00650
ANN with traingdx [6]	0.04300
EKF & C-means (Proposed) [23]	0.00310
PF & C-means (Proposed)	0.00240

Table 4: Comparison of S&P500 index prediction performance (MAPE) between the proposed algorithm and other algorithms

Algorithm	**MAPE**
HMM [11]	0.00703
EKF & C-means (proposed) [23]	0.00430
PF & C-means (proposed)	0.00360

Table 5: Comparison of prediction performance (MAPE) between for the proposed algorithm and other algorithms

Index Name	**ANN**	**ARIMA**	**Combination of HMM Fuzzy**	**MAP HMM Model**	**EKF & C-means (proposed) [23]**	**PF & C-means (proposed)**
Tata Steel [32]	--	--	--	1.560	1.173	1.105
Apple Inc. [32]	1.801	1.801	1.769	1.510	1.508	1.203
IBM Corp. [32]	0.972	0.972	0.779	0.611	0.478	0.437
Dell Inc. [32]	0.660	0.660	0.405	0.824	0.606	0.538

Table 6: Comparison of prediction performance between for the proposed algorithm and other algorithms

Index Name	**Perfor-mance Measure**	**KF& C-means**	**EKF& C-means**	**EKF& k-means**	**PF& C-means**	**PF& k-means**
S&P 500 [32]	MAPE	0.0055	0.0043	0.0050	0.0036	0.0041
MSFT [32]	MSE	0.0055	0.0031	0.0051	0.0024	0.0031
Tata Steel [32]	MAPE	1.658	1.173	1.553	1.105	1.2435
Apple [32]	MAPE	1.806	1.508	1.754	1.203	1.3535
IBM [32]	MAPE	0.654	0.478	0.611	0.437	0.4911
Dell [32]	MAPE	0.808	0.606	1.468	0.538	1.1796

Table 5 compares the prediction results of PF & C-means (proposed) with Hidden Markov Model (HMM) [9] and EKF & C-means model (proposed [23]) to predict close price for IBM, APPL, DELL, and TATA STEEL indices.

8. CONCLUSION

This article presents a novel way to predict future data point value of financial indices time series using C-means and Particle filter. A number of Benchmark datasets for financial indices (S&P 500, MSFT, IBM, DELL, APPL, Tata Steel) are used to evaluate the performance of the proposed algorithm. Experimental results proved the effectiveness and the superiority of the proposed algorithm based on the comparison of the results with those of other algorithms.

As for the future work, the proposed method could be used to predict future values in different fields of time series such as weather, enrolment etc. Also, the prediction of two future points or more could be implemented. Another future work may focus on using optimization algorithms to tune C-means and Particle filter parameters.

9. REFERENCES

[1] Brockwell, P. J. and Davis, R. A. Introduction to Time Series and Forecasting,3 th ed, Verlag New York: Springer, 2016, p. 449.Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[2] Samanta, B. "Prediction of chaotic time series using computational intelligence," Expert Systems with Applications, vol. 38, no. 9, pp. 11406-11411, 2011.Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[3] Sapkal, S., Barate, T., Kadbane, N., Pimple , M. and Kumbharkar, P. "Comparative Study and Analysis of Stock Market Prediction Algorithms," International Journal of Innovative Science, Engineering & Technology, vol. 3, March 2016.

[4] Adebiyi, A. A., Adewumi, A. O. and Ayo, C. K. "Stock Price Prediction Using the ARIMA Model," UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, pp. 106-112, 2014.

[5] Adebiyi, A. A, Adewumi, A. O and Ayo, C. K. "Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction," Journal of Applied Mathematics, vol. 2014, p. 7 pages, 2014.

[6] Abhishek, K., Khairwa, A. and Sur, T. P. "A Stock Market Prediction Model using Artificial Neural Network," International Conference on Computing, Communications and Networking Technologies (ICCCNT),IEEE-20180, 26 th -28 th July 2012.

[7] Somani, P., Talele, S. and Sawant, S. "Stock Market Prediction Using Hidden Markov Model," IEEE, 2014.

[8] LIN, Y., GUO, H. and HU, J. "An SVM-based Approach for Stock Market Trend Prediction," in Neural Networks (IJCNN), The 2013 International Joint Conference on.IEEE, pp. 1-7, 2013.

[9] Gupta, A. and Dhingra, B. "Stock Market Prediction Using Hidden Markov Models," Engineering and Systems (SCES) Students Conference on, IEEE, 2012.

[10] Gupta, A. and Sharma, S. D. "Clustering-Classification Based Prediction of Stock Market Future Prediction," (IJCSIT) International Journal of Computer Science and Information Technologies, vol. 5, no. 3, pp. 2806-2809, 2014.

[11] Nguyen, N. Thi. "Probabilistic Methods in Estimation and Prediction of Financial Models," Florida State University Libraries, PHD dissertation, p. 86, 2014.

[12] Liang-Ying, W. "A hyprid ANFIS model based on emprical model decomposition for stcok time series forecasting," Appl Soft Comput, 2016.

[13] Marzi, H., Haj Darwish, A. and Helfawi, H. "Training ANFIS using The Enhanced Bees Algorithm and Least Squares Estimation," Intelligent Automation & Soft Computing, 2016.

[14] Xu, Y. and Zhang, G. "Application of Kalman Filter in the Prediction of Stock Price," International Symposium on Knowledge Acquisition and Modeling (KAM 2015), 2015.

[15] Taylor, S. J. Modeling financial time series, 2nd ed., World Scientific Publishing, 2007, p. 297.

[16] Adhikari, R. and Agrawal, R. K. "An Introductory Study on Time Series Modeling and Forecasting," LAP Lambert Academic Publishing, p. 67, 26 Feb 2013.

[17] Chatfield, C. The Analysis of Time Series: An Introduction, 6 th ed., 2016, p. 352 pages.

[18] Jafar, O. A. M. and Sivakumar, R. "A Comparative Study of Hard and Fuzzy Data Clustering Algorithms with Cluster Validity Indices," Proceeding of International Conference on Emerging Research in Computing, Information, Communication and Applications, 2013.

[19] Velmurugan, T. "Performance Comparison between k-Means and Fuzzy C-Means Algorithms using Arbitrary Data Points," Wulfenia Journal, vol. 19, pp. 234-241, 2012.

[20] Lu, Y., Ma, T., Yin, C., Xie, X., Tian, W. and Zhong, S. "Implementation of the Fuzzy C-Means Clustering Algorithm in Meteorological Data," International Journal of Database Theory and Application, vol. 6, pp. 1-18, 2013.

[21] Kannan, S., Ramathilagam, S. and Chung, P. "Effective fuzzy c-means clustering algorithms for data clustering problems," Expert Systems with Applications, p. 6292–6300, 2012.

[22] Chadaporn, K., Baber, J. and Bakhtyar, M. "Simple Example of Applying Extended Kalman Filter," 1st International Electrical Engineering Congress (iEEECON2013), March 2014.

[23] Hilal, A. and Haj Darwish, A. "A Novel Method for Stock Price Prediction based on Fuzzy Clustering and Extended Kalman Filter," Res. J. of Aleppo Univ., Engineering Sciences Series (2), no. 136, 2017.

[24] De Bernardis, C., Vicente-Guijalba, F., Martinez-Marin, T. and Lopez-Sanchez, J. M. "Particle Filter Approach for Real-Time Estimation of Crop Phenological States Using Time Series of NDVI Images," Remote Sensing, 20 July 2016.

[25] Arulampalam, M. S., Maskell, S., Gordon, N. and Clapp, T. "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," IEEE TRANSACTIONS ON SIGNAL PROCESSING, vol. 50, FEBRUARY 2002.

[26] Murphy, J. "Bayesian methods for high frequency financial time series analysis," Cambridge Univ. Dept. of Eng., Cambridge, U.K, 2010.

[27] Yin, S. "Intelligent Particle Filter and Its Application on Fault Detection of Nonlinear System," IEEE Transactions on Industrial Electronics, p. 10, 2015.

[28] Capitaine, H. L. and Frelicot, C. "A cluster validity index combining an overlap measure and a separation measure based on fuzzy aggregation operators," IEEE Transactions on Fuzzy Systems, July 2011.

[29] Trujillo-Ortiz, A. and Hernández Walls, R. "Mskekur: Mardia's multivariate skewness and kurtosis coefficients and its hypotheses testing," 2003. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=3519. [Accessed 13/ 9/ 2017].

[30] Trujillo-Ortiz, A., Hernández Walls, R. and Barba-Rojo, K. "HZmvntest: Henze-Zirkler's Multivariate Normality Test," 2007. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do? objectId=17931. [Accessed 13/ 9/ 2017].

[31] Trujillo-Ortiz, A., Hernández Walls, R., Barba-Rojo, K. and Cupul-Magana, L. "Roystest: Royston's Multivariate Normality Test," 2007. [Online]. Available: http://www.mathwoks.com/matlabcentral/fileexchange/17811. [Accessed 13/ 9/ 2017].

[32] "YAHOO FINANCE," [Online]. Available: www.yahoofinanace.com. [Accessed 8/ 2017].