CS-403 PROJECT REPORT

On

Forecasting financial series using clustering methods and support vector regression

Submitted by:

Rahul Kumar	**190001049**
Harsh Vardhan Agrawal	**190005016**
Vivek Bhushan	**190005043**

Submitted to:

Prof. Kapil Ahuja
(Professor, CSE)

DISCIPLINE OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, INDORE

April, 2023

Index

1 Abstract

This work aims to reproduce and extend a portion of the original paper titled "Forecasting financial series using clustering methods and support vector regression." Specifically, we focus on implementing the K-means clustering algorithm in conjunction with Support Vector Regression (SVR) for forecasting financial time series. Our objective is to compare the performance of our proposed model with the results reported in the original paper.

To achieve this, we employ a two-stage approach similar to the original study. In the first stage, we utilize K-means clustering to segment the financial time series into distinct contexts or clusters. This segmentation allows us to capture the underlying patterns and dynamics present in the data more effectively. In the second stage, we employ SVR models, with each cluster corresponding to a separate SVR, to forecast future values of the series.

We conduct our experiments using a financial series comprising values from an equity fund of a Brazilian bank, as done in the original paper. To evaluate the performance of our proposed model, we compare it to the hierarchical model (HM) presented in the literature, which had demonstrated superior predictive results compared to Support Vector Machine (SVM) and Multilayer Perceptron (MLP) models in the original work.

Preliminary results show promising outcomes, indicating that our proposed model outperforms the HM from the original paper. Specifically, we achieve a 32% reduction in forecasting error compared to the HM. This improvement suggests that our model is also superior to the SVM and MLP models examined in the original study.

Additionally, we analyze the construction and utilization of clusters in relation to the volatility of the series. Our findings indicate that data obtained from either low or high

volatility alone provide sufficient knowledge to the model, enabling accurate forecasts of future values. Moreover, our analysis of the quality of the formed clusters reveals that each cluster carries distinct information about the series. Furthermore, we observe the presence of a group of SVRs capable of making accurate forecasts, with the majority of forecasts relying on an SVR belonging to this group.

In conclusion, our work replicates and expands upon a portion of the original paper by implementing K-means clustering with SVR for financial time series forecasting. Our results demonstrate the superiority of our proposed model over the HM model, SVM, and MLP models. The findings also shed light on the importance of considering volatility and the informative nature of clusters in improving the accuracy of financial forecasts.

2 Introduction

2.1 About Financial Time Series

A time series represents a sequential collection of observed values or data points that occur at regular intervals, such as hourly, daily, weekly, or yearly. Forecasting future values of a time series is widely utilized across various domains to aid in planning and decision-making processes. The ease or complexity of forecasting can be influenced by several factors. These factors include our understanding of elements that impact the behavior of the underlying event generating the time series, the amount of information available within the series, and the extent to which previous forecasts can affect future predictions of the series.

Financial time series are challenging to model and forecast due to their inherent characteristics of noise and non-stationarity. Noise refers to the lack of information within the series, making it difficult to establish accurate relationships between past and future observations. Non-stationarity indicates that the statistical distribution of the series changes over time. Introducing more information into the series can help reduce the impact of noise but may lead to increased non-stationarity. These factors pose significant challenges in effectively analyzing and predicting financial time series.

2.2 Previous Research Work

Machine learning models have gained significant popularity in the analysis and forecasting of financial time series. Researchers have explored various techniques and models to tackle this task. For example, dimensionality reduction techniques and artificial neural network (ANN) models have been tested to forecast values in financial time series. Additionally, studies have compared the classification performance of ANN and support vector machines (SVMs) in predicting financial time series. Some researchers have even developed trading systems based on rough set analysis.

Numerous papers have been published on the utilization of machine learning models in financial time series analysis and forecasting, indicating the widespread adoption of these approaches in the field.

2.3 Current Work

This paper proposes a two-stage model for forecasting financial time series. In the first stage, clustering methods such as K-Means or Fuzzy C-Means are employed to segment the time series into distinct contexts or clusters. In the second stage, support vector regressions (SVRs) are utilized, with one SVR assigned to each cluster, to predict future values of the series.

The model operates by evaluating the relevance of a given pattern to the identified clusters. Based on this assessment, the model selects the most pertinent cluster and utilizes the corresponding SVR to generate a forecast. Two forecasting processes are employed. The first process involves using the SVR associated with the cluster most relevant to the pattern to predict the future value. The second process combines the forecasts of SVRs from all clusters, weighting each forecast according to the membership of the pattern in relation to each cluster.

This two-stage model leverages clustering to capture the contextual information within the financial time series and utilizes SVRs for accurate forecasting based on the identified clusters.

The experiments conducted in this paper utilize a financial time series comprising values from an equity fund. The proposed model is compared to a hierarchical model (HM) introduced in a previous study. The HM consists of a segmentation step using a Self-Organizing Map (SOM) and a forecasting step using a Support Vector Machine (SVM).

In the previous study, the HM outperformed both an SVM model and a multilayer perceptron (MLP) model in terms of forecast accuracy on the same financial time

series. Therefore, by comparing the proposed model to the HM, we indirectly evaluate its performance against SVM and MLP models as well.

This comparison aims to assess the effectiveness of the proposed model in forecasting financial time series and provides insights into its relative performance compared to the HM, SVM, and MLP models.

3. Brief Review of Theory Involved

3.1 Clustering Methods

3.1.1 K-Means Clustering

The K-Means clustering method, introduced by MacQueen in 1967, is a technique used to partition a dataset containing n objects into k distinct and non-overlapping clusters. However, the concept of K-Means clustering can be extended to other clustering algorithms as well. The primary objective of K-Means clustering, and similar methods, is to assign each object to a specific cluster based on its distance from the centroid of that cluster.

In the context of K-Means clustering, the centroid (c_i) of a cluster represents the average position of all the objects that belong to that cluster. It is calculated using Equation 1, where $u_{ij} \in \{0, 1\}$ denotes the membership of the object x_j in cluster i.

The formula to calculate the centroid (c_i) is not explicitly provided here, but it involves summing the values of each object belonging to the cluster and dividing the sum by the total number of objects in that cluster. The centroid serves as a representative point within the cluster and is used to measure the similarity or dissimilarity between objects in different clusters.

Overall, K-Means clustering and similar methods utilize the concept of centroids to determine the membership of each object in a cluster based on their distances from the cluster's centroid. This allows for the creation of distinct clusters that group similar objects together.

$$c_i = \frac{\sum_{j=1}^{n} u_{ij} x_j}{\sum_{j=1}^{n} u_{ij}}$$

The basic rules of the K-Means method are:

1. Empty clusters are not allowed.

2. An object must belong to a single cluster.
3. The union of all clusters forms the original set of objects.

3.1.2 Fuzzy C-Means

The Fuzzy C-Means (FCM) method is a soft clustering algorithm that allows objects to have partial membership in different clusters. Unlike hard clustering methods, where objects are assigned exclusively to a single cluster, FCM assigns degrees of membership to objects in each cluster, ranging from 0 to 1.

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^{m} x_j}{\sum_{j=1}^{n} u_{ij}^{m}}$$

The basic steps of the Fuzzy C-Means method are as follows:

1. Initialization: Determine the number of clusters (k) and randomly assign initial membership values (u_{ij}) for each object, satisfying the condition $0 \leq u_{ij} \leq 1$.
2. Calculate Cluster Centers: Compute the centroid (c_i) for each cluster based on the membership values. The centroid is calculated as a weighted average of object positions, where the degree of membership serves as the weight.
3. Update Membership Values: Recalculate the membership values (u_{ij}) for each object based on their distances from the cluster centroids. Objects closer to a particular centroid will have a higher membership value for that cluster.
4. Repeat Steps 2 and 3: Iterate Steps 2 and 3 until convergence criteria are met. Convergence is typically determined by a predefined threshold or when the membership values no longer change significantly.

The Fuzzy C-Means method follows some basic rules during the iterations:

1. Membership Calculation: The membership value (u_{ij}) for each object should be updated based on its distance from the cluster centroids. The distance can be computed using various distance metrics such as Euclidean distance.
2. Cluster Centers Update: After updating the membership values, the centroids (c_i) of the clusters are recalculated by considering the new membership values. The centroid is calculated as the weighted average of the object positions, where the weights are the membership values.
3. Membership Constraints: The membership values for each object in each cluster must satisfy the condition $0 \leq u_{ij} \leq 1$. Additionally, the sum of membership values for an object across all clusters should equal 1, ensuring that each object's membership is fully accounted for.
4. Convergence Criterion: The algorithm continues iterating until a convergence criterion is met. This criterion can be defined based on a maximum number of iterations or when the changes in membership values or centroids become negligible.

By allowing partial membership, the Fuzzy C-Means method provides a more nuanced and flexible clustering approach compared to hard clustering methods, accommodating situations where objects may have similarities with multiple clusters.

3.2 SVM and SVR

Support Vector Machine (SVM) is a powerful machine learning algorithm initially introduced by Boser et al. in 1992 and further detailed by Vapnik in 1998. It is commonly used for pattern classification tasks and aims to construct a hyperplane that maximally separates positive and negative patterns.

In SVM, the goal is to find a decision boundary or hyperplane that best separates the data points into different classes. The hyperplane is determined by a subset of training samples called support vectors, which are the closest points to the decision boundary.

For binary classification, where there are two classes (+1 and -1), the SVM seeks to find a hyperplane represented by the equation:

$$w \cdot x - b = 0$$

Here, w is the weight vector normal to the hyperplane, x represents the input feature vector, and b is the bias term. The sign of $w \cdot x - b$ determines the predicted class label.

To maximize the margin between the hyperplane and the support vectors, SVM introduces the concept of "soft margin." It allows for some degree of misclassification by introducing slack variables (ξ) to handle the cases where points are not perfectly separable.

The objective of SVM is to minimize the following optimization problem:

$$\min \tfrac{1}{2}||w||^2 + C\Sigma\xi$$

Subject to:

$$y_i(w \cdot x_i - b) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

Here, y_i represents the class label for the ith training sample, C is a regularization parameter controlling the trade-off between maximizing the margin and minimizing the misclassification, and ξ_i represents the slack variables.

Support Vector Regression (SVR) extends the concept of SVM to handle regression tasks. Instead of finding a hyperplane for classification, SVR aims to find a hyperplane that best fits the data points within a specified margin or tube.

The objective of SVR is to minimize the following optimization problem:

$$\min \tfrac{1}{2}||w||^2 + C \Sigma \xi_i + \varepsilon\Sigma(\xi_i + \xi_i^*)$$

Subject to:

$$y_i - w \cdot x_i - b \leq \varepsilon + \xi_i$$

$$w \cdot x_i + b - y_i \leq \varepsilon + \xi^*_i$$

$$\xi_i \ , \xi^*_i \geq 0$$

Here, y_i represents the target value of the ith training sample, ε is the tube width determining the insensitive region, ξ_i and ξ^*_i are slack variables, and C controls the trade-off between fitting the data and allowing for errors.

In both SVM and SVR, the optimization problem is typically solved using techniques such as quadratic programming to find the optimal hyperplane that maximizes the margin or best fits the data points.

4 Feature Analysis

4.1 Problem with the Series

As previously stated, a time series refers to a collection of observations arranged in chronological order. Time series data often exhibit unpredictable patterns or randomness. While it is not possible to accurately predict their future values with certainty, they do demonstrate a specific form of probabilistic behavior.
The problem lies in the inherent uncertainty associated with time series forecasting. Due to the random nature of the data, traditional deterministic methods cannot provide precise predictions for future values. Instead, time series forecasting relies on statistical and probabilistic approaches to capture the underlying patterns and trends within the data.
The challenge is to develop models and techniques that can effectively capture and analyze the stochastic behavior of time series data. By understanding the patterns and relationships within the data, we can make informed predictions and estimate the likely range of future values. This requires employing advanced forecasting methods such as statistical models, machine learning algorithms, or time series analysis techniques.
The aim is to strike a balance between acknowledging the random nature of time series data and uncovering meaningful patterns that allow us to make reasonable forecasts. By leveraging the inherent structure and statistical properties of time series, we can improve our understanding of their behavior and enhance the accuracy of predictions.

4.2 Calculating Continuous Compound Series

Continuous compound return series, also known as return time series, are commonly used in analyzing the behavior of asset prices. These series possess certain

characteristics that make them suitable for studying asset performance. For instance, return time series typically do not exhibit long-term trends or scaling components, and they are considered weakly stationary.

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right) = \ln(P_t) - \ln(P_{t-1})$$

A stationary time series is one in which the statistical properties, such as the mean and variance, remain constant over time. In the case of weak stationarity, while the assumption of strict equality in probability distribution is relaxed, the mean and autocovariance between different time points are still considered constant.

In the context of return time series, weak stationarity implies that the average return remains constant over time, indicating no significant long-term upward or downward trend. Additionally, the autocovariance, which measures the linear dependence between the returns at different time lags, remains consistent. This means that the relationship between the return at time t and the return at a lagged time t_l remains stable for any arbitrary integer l.

The concept of continuous compound return refers to the calculation of returns over successive periods. Instead of considering simple returns (price differences), continuous compound returns are computed by taking the logarithm of the ratio of prices between two time points. This approach allows for the aggregation of returns over multiple periods, yielding a cumulative measure of return.

Continuous compound return series are often preferred in financial analysis because they account for the compounding effect of returns. By employing logarithmic transformations, these series help capture the proportional growth or decline in asset prices more accurately. This is particularly relevant when studying investments over longer periods or comparing returns across different assets.

Overall, continuous compound return series offer desirable features for analyzing asset price behavior due to their tendency to be weakly stationary, lack of long-term trends, and ability to capture the compounding nature of returns. These properties make them

valuable in various financial analyses, including risk assessment, portfolio optimization, and forecasting future asset prices.

4.3 Auto-Covariance and Correlogram Mapping

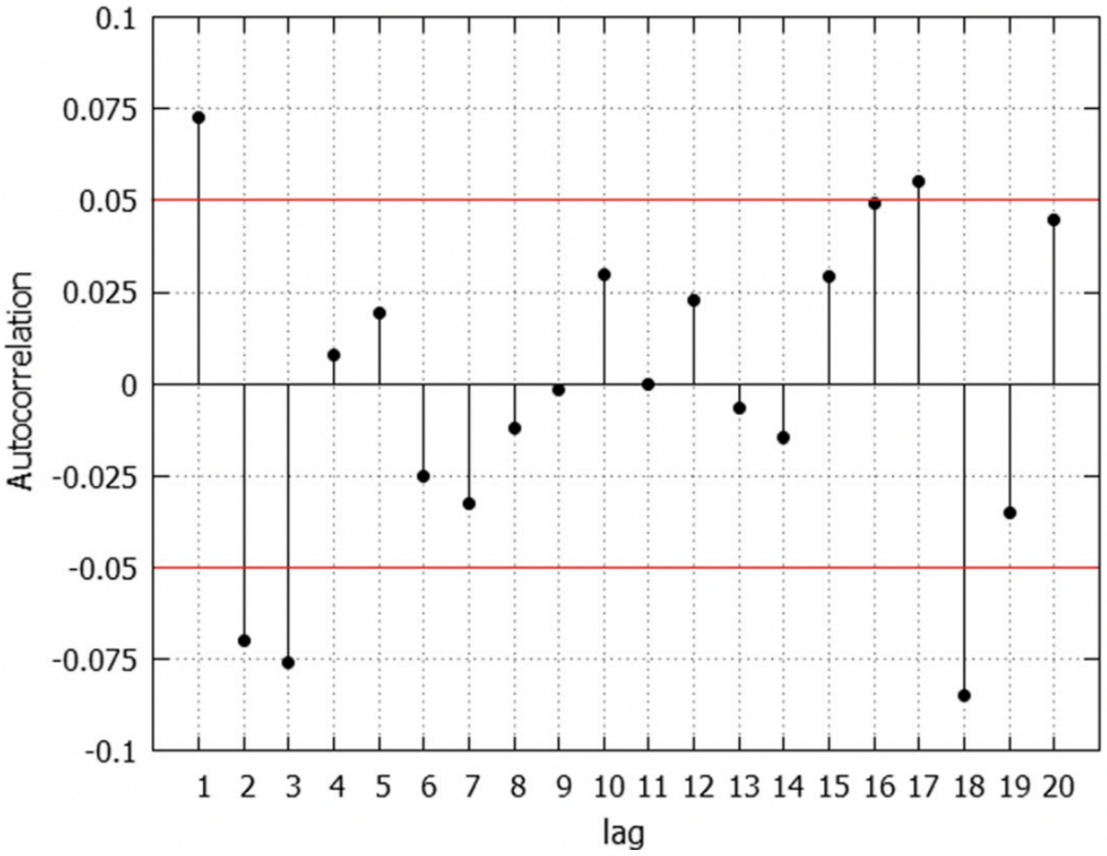
In the context of time series analysis, the concept of stationarity is important in understanding the behavior and properties of the data. Stationarity refers to a condition where the statistical properties of a time series remain constant over time.
In a strictly stationary time series, the probability distribution of the random variables that make up the series remains the same across all time points. This implies that the mean, variance, and other statistical properties are constant over time. Consequently, any relationship or correlation observed between different time points is considered to be stable and consistent.

On the other hand, weak stationarity, also known as covariance stationarity, relaxes the strict assumption of identical probability distributions. In a weakly stationary series, while the exact distribution of the random variables may vary, the mean and autocovariance remain constant over time.

To calculate autocovariance in the context of a continuous compound return series, we consider the returns at different time lags. Autocovariance measures the covariance between a return at a particular time t and a return at a lagged time t-l, where l is an arbitrary integer representing the time lag.

By calculating the autocovariance function for a weakly stationary series, we can determine the relationship or dependence between the returns at different time lags. The magnitude and sign of the autocovariance provide insights into the strength and direction of the linear relationship between the current and previous returns.

The autocovariance values can then be used to compute the autocorrelation, which measures the linear relationship between the returns at different lags. Autocorrelation helps us understand the persistence or predictability of returns over time. A positive autocorrelation indicates that returns tend to follow similar patterns or trends over successive periods, while a negative autocorrelation suggests an inverse relationship.



By examining the autocovariance and autocorrelation of a continuous compound return series, we can gain insights into the temporal dependencies and patterns within the data. This information is valuable for understanding the behavior of asset prices, identifying trends, and developing forecasting models.

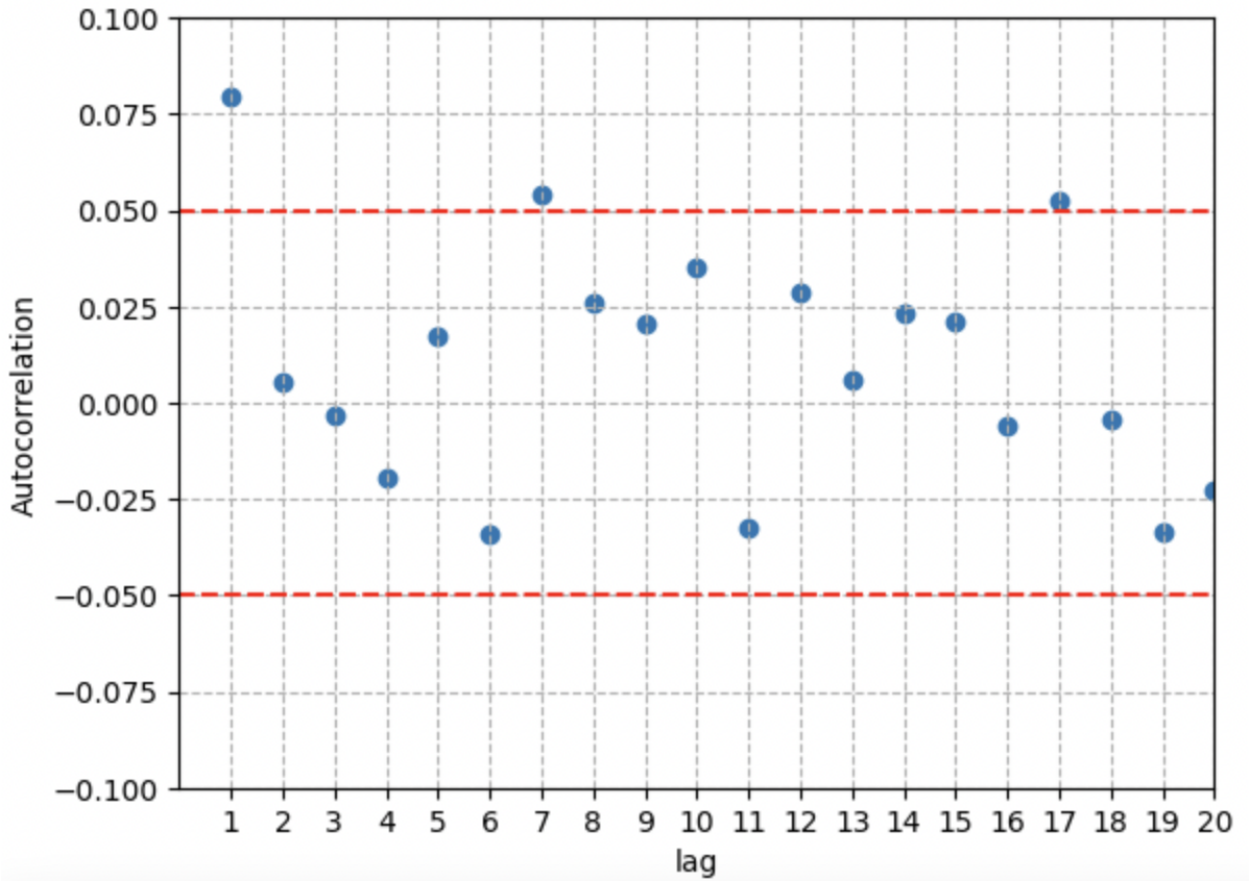
In this particular case, the correlogram analysis revealed that there are three past values, denoted as l_1, l_2, and l_3, that exhibit substantial correlation with the subsequent

value in the series. These values are often referred to as r_{t+f-1}, r_{t+f-2}, and r_{t+f-3}, where r_{t+f} represents the future value being forecasted.

Based on the high correlation observed for these specific lagged values, they were chosen as features for the forecasting model. By including these relevant lagged values as features, the complexity of the forecast is reduced since they capture important information related to the future value.

It's worth noting that while there may be other lagged values, such as l_{17} and l_{18}, which appear to have significant correlations, they are considered spurious in the time series modeling. Spurious correlations are those that occur due to random chance rather than meaningful relationships. Therefore, these higher-lagged values are typically disregarded in the modeling process.

By selecting the lagged values with the highest and most significant correlations, the forecasting model can effectively capture the temporal dependencies and patterns within the time series, leading to more accurate predictions.



Plot Made on Our Dataset

4.4 Volatality in Time Series

In the context of stock price prediction, volatility is an important feature to consider alongside lagged values of the stock price. Volatility refers to the degree of variation or fluctuation in the price of a financial asset over a specific period.

Including volatility as a feature in the prediction model is valuable because it captures the inherent uncertainty and risk associated with the market. Here are a few reasons why volatility is important in market prediction models:

1. Impact on Returns: Volatility affects the potential returns of a stock. Higher volatility implies larger price swings, which can lead to both higher potential

gains and losses. Investors and traders often consider volatility when assessing the risk-reward profile of a particular stock or market.

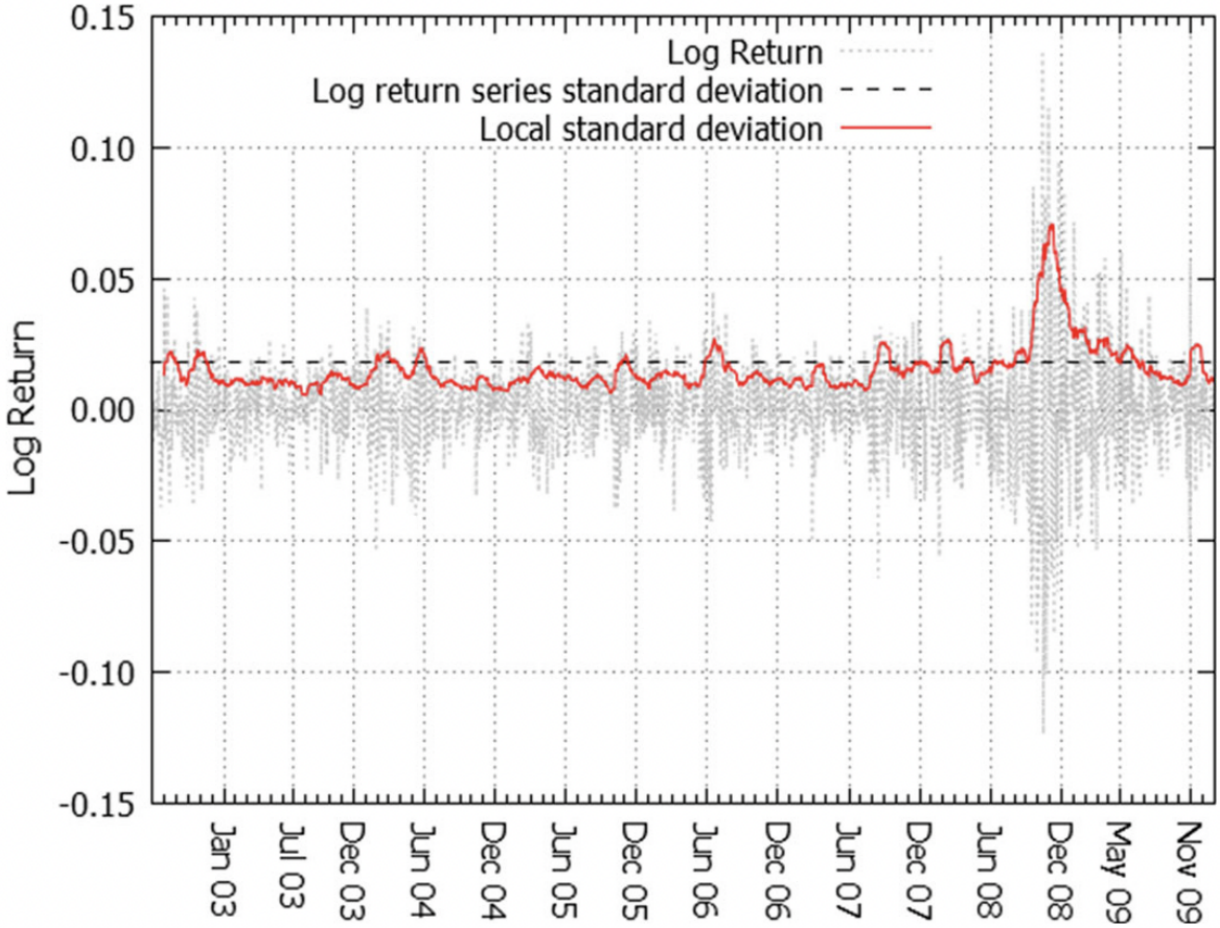
2. Market Trend Identification: As mentioned in the passage, uptrend periods typically exhibit lower volatility compared to downtrend periods. Volatility can provide insights into the overall market trend, helping to identify whether the market is in a bullish (rising prices) or bearish (falling prices) phase. Incorporating volatility as a feature can help the prediction model capture the changing market dynamics.
3. Risk Management: Volatility is closely linked to risk management. Higher volatility suggests increased uncertainty and potential market instability. By considering volatility as a feature, the prediction model can account for market conditions that may impact the stock price, enabling better risk assessment and management.
4. Trading Strategies: Volatility plays a significant role in developing trading strategies. Some traders prefer high volatility environments as they offer more significant profit opportunities through price fluctuations. On the other hand, low volatility periods may require different trading approaches to account for reduced price movements. Including volatility as a feature allows the model to adapt its predictions based on the prevailing market conditions.

By incorporating volatility as an additional feature alongside lagged values of the stock price, the prediction model can capture the relationship between market dynamics, price movements, and risk. This helps improve the accuracy and robustness of the model in forecasting stock prices and making informed investment decisions.

To incorporate volatility as a binary feature in the stock price prediction model, a strategy based on the available data is employed. The return series is divided into periods of high and low volatility. The volatility of a particular value, denoted as si, is determined by comparing the standard deviation of the entire return series with the local standard deviation.

The local standard deviation is calculated as the standard deviation of the range of the twenty values preceding si in the return series. If the local standard deviation is greater than the standard deviation of the entire series, si is considered to have high volatility.

Conversely, if the local standard deviation is smaller, si is regarded as having low volatility.

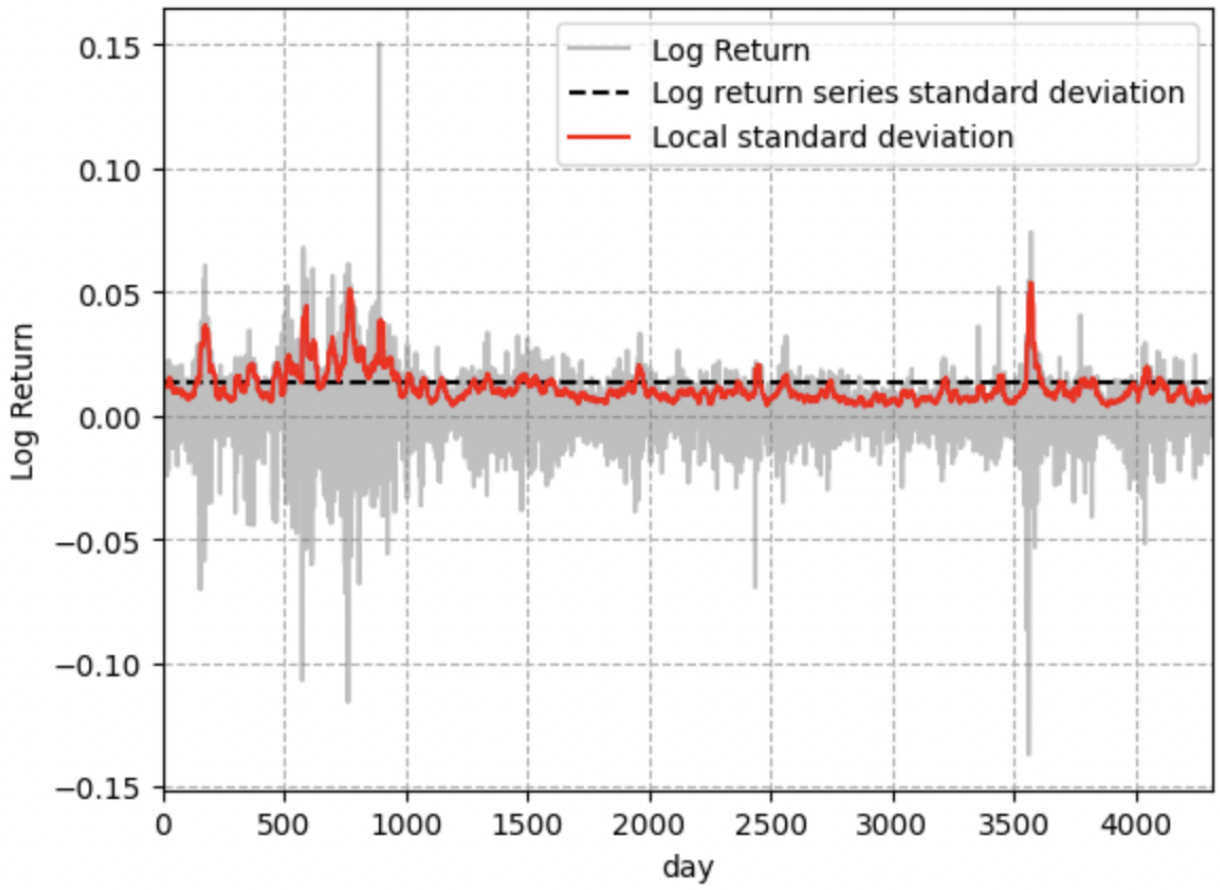


Using the standard deviation as a measurement for determining volatility is appropriate because it provides information about the historical dispersion of the values in the series relative to their mean. Higher standard deviation indicates greater variability and

potential for larger price fluctuations, indicating high volatility. Conversely, lower standard deviation suggests relatively stable and less volatile market conditions.

By applying this strategy, the return series is divided into distinct periods of high and low volatility. This binary representation of volatility can serve as an additional feature in the stock price prediction model. It enables the model to capture the varying market dynamics and adjust its predictions accordingly based on the prevailing level of volatility.

Incorporating binary volatility values as features in the model enhances its ability to account for different market conditions and potentially improve the accuracy of stock price predictions.



Plot Made on Our Dataset

4.5 Final Feature Matrix

Pattern (condition attributes)				Decision attribute
r_{t+f-3}	r_{t+f-2}	r_{t+f-1}	Vol	r_{t+f}

To construct the final decision matrix for training and testing the stock price prediction model, the return series values are organized as patterns in a decision table. Let's assume we have a sample of n patterns in the decision table. The decision table consists of the following attributes:

1. Condition Attributes:
    - r_{t+f-1}: The lagged value of the return series one day before the target value.
    - r_{t+f-2}: The lagged value of the return series two days before the target value.
    - r_{t+f-3}: The lagged value of the return series three days before the target value.
    - Volatility: A binary value representing the volatility of the target value. A value of 0 indicates low volatility, while a value of 1 represents high volatility.
2. Decision Attribute:
    - r_{t+f}: The future value to be forecasted in the return time series.

Each pattern in the decision matrix represents a specific combination of the lagged values and volatility. The "x" denotes the corresponding values in the return series for each attribute. The decision attribute, r_{t+f} represents the target value to be predicted.

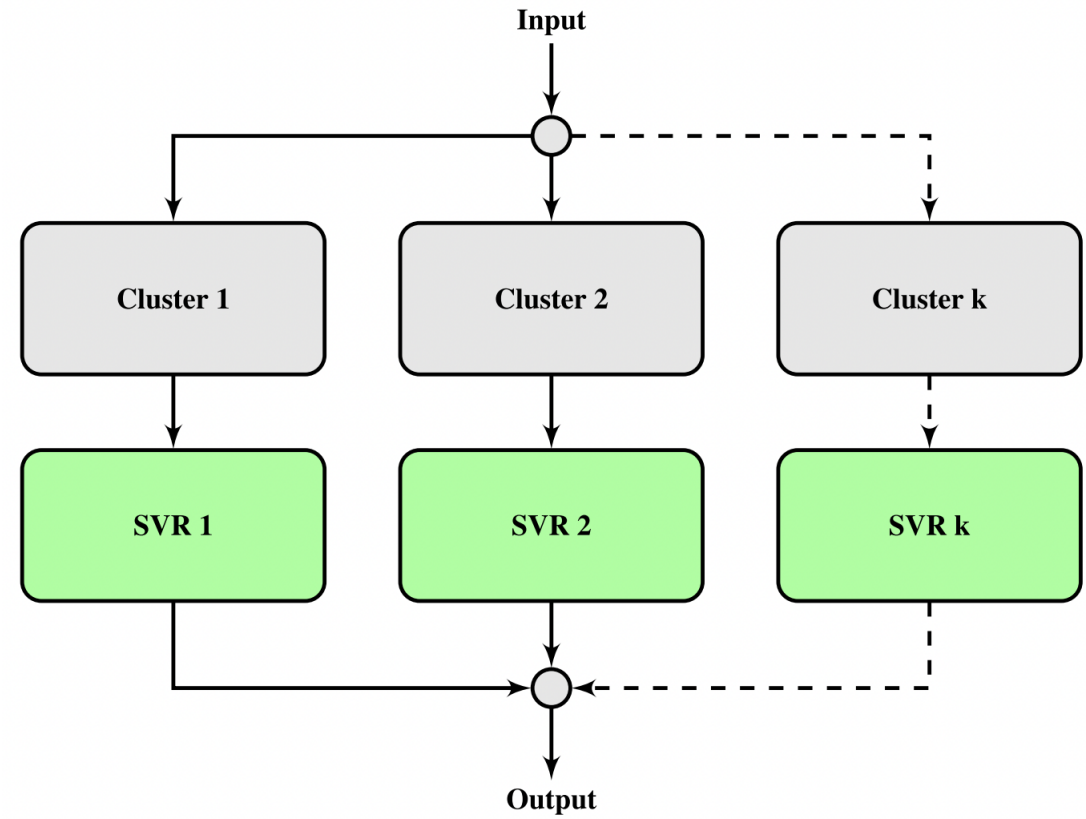
By using this decision matrix, we can train and test our stock price prediction model. The model will learn the relationship between the condition attributes (lagged values and volatility) and the decision attribute (future value) to make accurate predictions of the target stock price.

5 Model Training & Prediction

5.1 Architecture of Model

The training process of the model consists of two phases: clustering and modeling the time series behavior within each cluster. This approach differs from previous studies where SVM or hierarchical models were trained on the entire dataset without clustering.

In the first phase, the patterns in the decision table are segmented into clusters. This is achieved by executing either the K-Means method or the Fuzzy C-Means method on the patterns. These methods use the values of the return series as inputs and group similar patterns together based on their similarity. The similarity measure used in both methods is the Euclidean distance, which calculates the distance between patterns. Patterns that are closer to each other (i.e., more similar) are assigned to the same cluster.



After the patterns have been segmented into clusters, the second training phase involves modeling the time series behavior within each cluster. This is done by applying support vector regression (SVR) separately to each cluster. SVR is a machine learning technique that constructs a regression model to predict future values based on past observations. By modeling the behavior within each cluster, the model can capture the specific characteristics and trends unique to that cluster.

Overall, this two-phase training architecture allows for a more fine-grained analysis of the time series data. By dividing the dataset into clusters and modeling each cluster individually, the model can better capture and understand the variations and patterns within the data, leading to more accurate predictions of future values.

5.2 Prediction

The proposed model takes input patterns in the format of the decision table and provides a forecast for the next value of the return series. The prediction phase of the model involves two stages: determining the membership of the input pattern in relation to the clusters and using support vector regressions (SVRs) to predict the next value.

In the first stage, the model determines the membership of the input pattern in each of the clusters. This membership represents the degree of similarity between the pattern and each cluster. The cluster with the highest membership is considered the most relevant for the input pattern.

In the second stage, there are two approaches that can be followed.

- The first approach is to use a single SVR that is specifically trained on patterns similar to the input pattern. If hard clustering methods are used, the pattern's membership degree will be total for a single cluster and zero for others. If soft

clustering methods are used, only the cluster with the highest membership degree will be chosen. The selected SVR, in this case, is the most specialized one to predict the future value because it has been trained exclusively on similar patterns.

- The second approach involves using all SVRs in the model to generate individual forecast values. Each SVR produces its own forecast based on its training. The membership degrees of the input pattern with respect to the clusters are used as weights to generate a final forecast value. Soft clustering methods are suitable for this approach as they provide membership degrees. SVRs trained on patterns that are more similar to the input pattern have a greater influence on the final predicted value compared to those trained on dissimilar patterns.

The second approach is similar to the hierarchical model (HM) proposed by where all contexts are used to produce the forecast value. It combines the forecasts from multiple SVRs, with each SVR specialized in predicting values within a specific cluster. Overall, the prediction phase combines clustering techniques to determine the relevant cluster for the input pattern and then utilizes SVRs to generate forecasts. The choice between using a single SVR or combining forecasts from all SVRs depends on the clustering method employed and the degree of membership of the input pattern in the clusters.

6 Experiments & Testing

The original research conducted six experiments, each focused on forecasting a specific period of time in the return series. Each period consisted of twenty consecutive days, which roughly corresponds to a one-month trading period. This duration takes into account the closure of the stock market during weekends and holidays. The experiments aimed to assess the performance of the proposed model in forecasting the return series over these specific time intervals.

Experiment	Time period		Volatility	Trend
	Start	End		
1	2004-11-22	2004-12-17	Low	Uptrend
2	2007-05-28	2007-06-25	Low	Uptrend
3	2009-09-11	2009-10-08	Low	Uptrend
4	2006-05-19	2006-06-16	High	Correction
5	2008-01-18	2008-02-18	High	Horizontal
6	2009-10-14	2009-11-11	High	Horizontal

In the experiment, six different time periods were selected to evaluate the performance of the proposed model. These time periods are described in Table 2. The first three periods are characterized by low volatility, while the last three periods exhibit high volatility.

Each time period represents a specific context within the overall time series. The first three periods are part of a clear uptrend movement. The fourth period corresponds to a correction (downward) movement following a previous uptrend. The fifth period represents a horizontal movement within an uptrend, and the sixth period represents a long-term horizontal movement.

These time periods were intentionally chosen to match the periods used in a previous study by Carpinteiro et al. (2012). This allows for a direct comparison of the results

obtained by the proposed model with those obtained by the Hierarchical Model (HM) and other models discussed in that study.

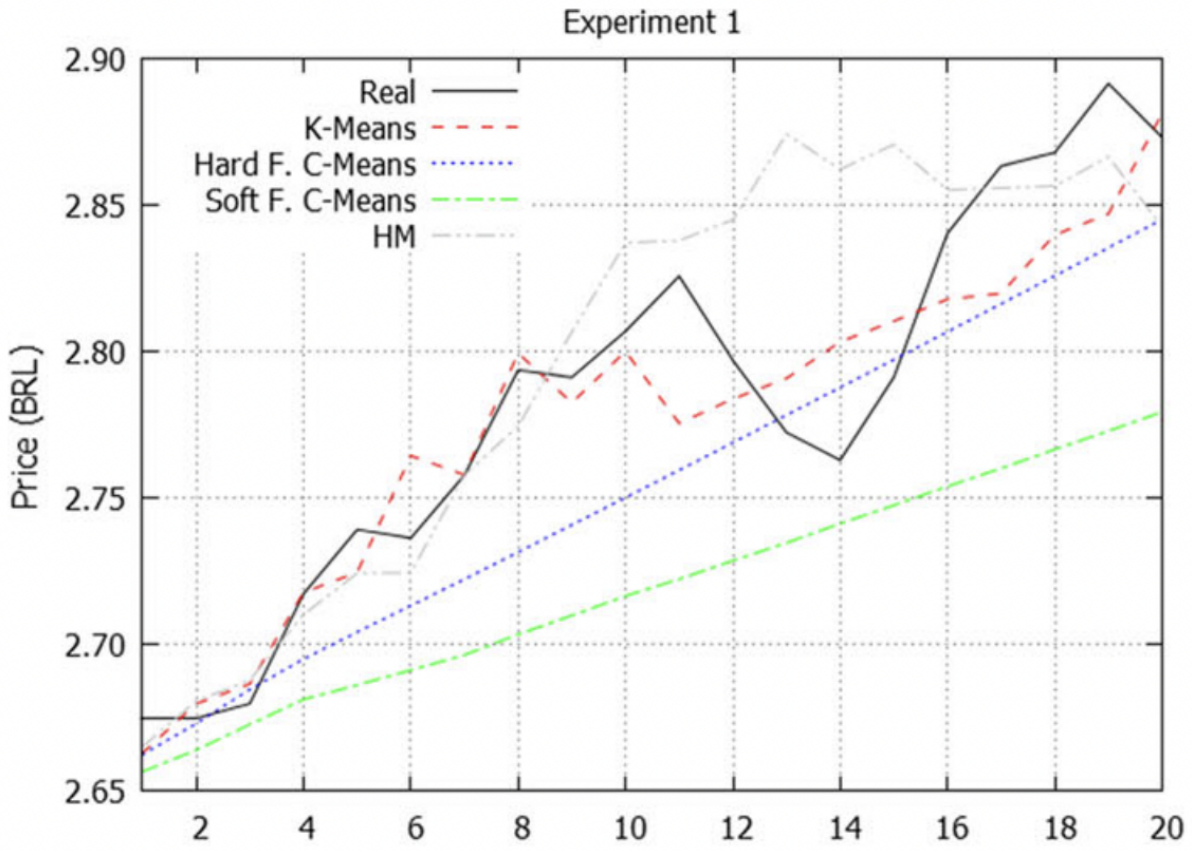
During each experiment, the values of the time period being forecasted were removed from the return series. This was done both for forming the clusters in the first stage and training the Support Vector Regressors (SVRs) in the second stage. The forecasting process was performed one step ahead, where the forecasted value was used as input for generating the next forecasted value. This iterative process continued until all twenty days in the time period were forecasted.

Both clustering methods, K-Means and Fuzzy C-Means, were utilized in each experiment. The Fuzzy C-Means method employed both the single SVR approach and the weighted composition approach described earlier. On the other hand, the K-Means method only utilized the single SVR approach since it is a hard clustering method. The SVRs used the radial basis function as the kernel function, as it yielded the best results in the study conducted by Carpinteiro et al. (2012).

7 Results

7.1 Original Paper

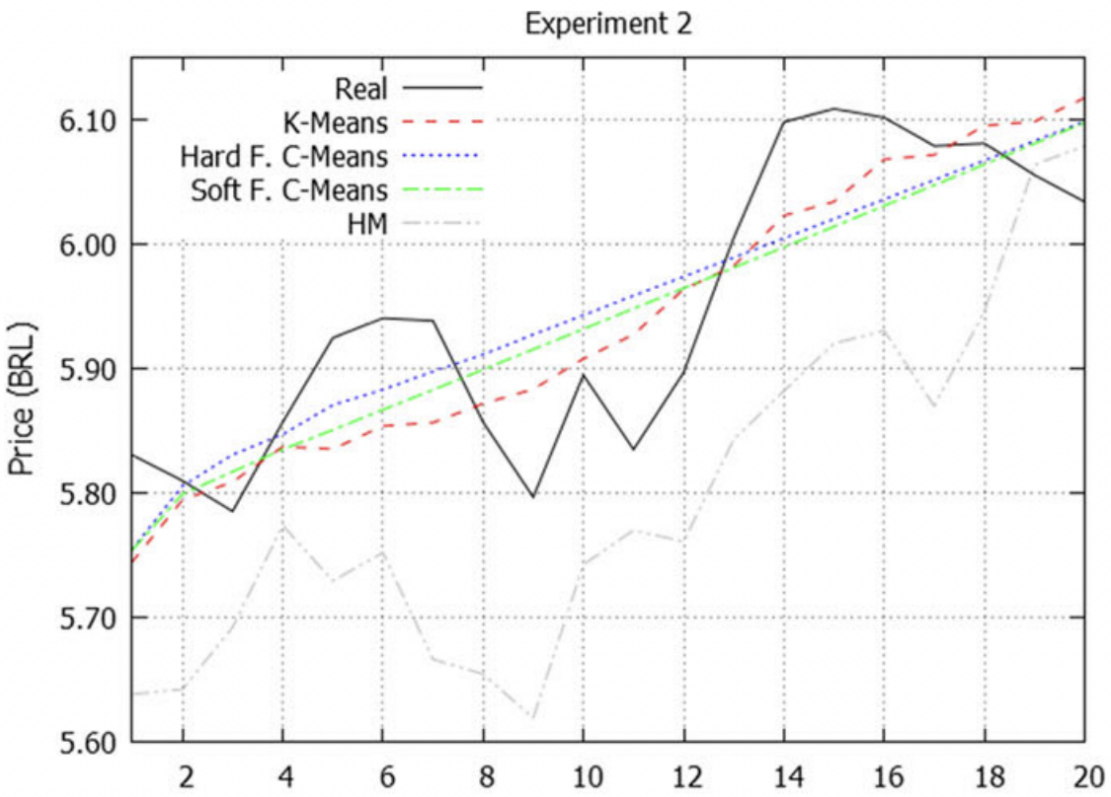
The graphical analysis presented here compares the performance of different models and their ability to forecast the stock price series. The SVM, MLP, and CRB models are not considered in this analysis because they produced poorer results compared to the other models.



Experiment 1:

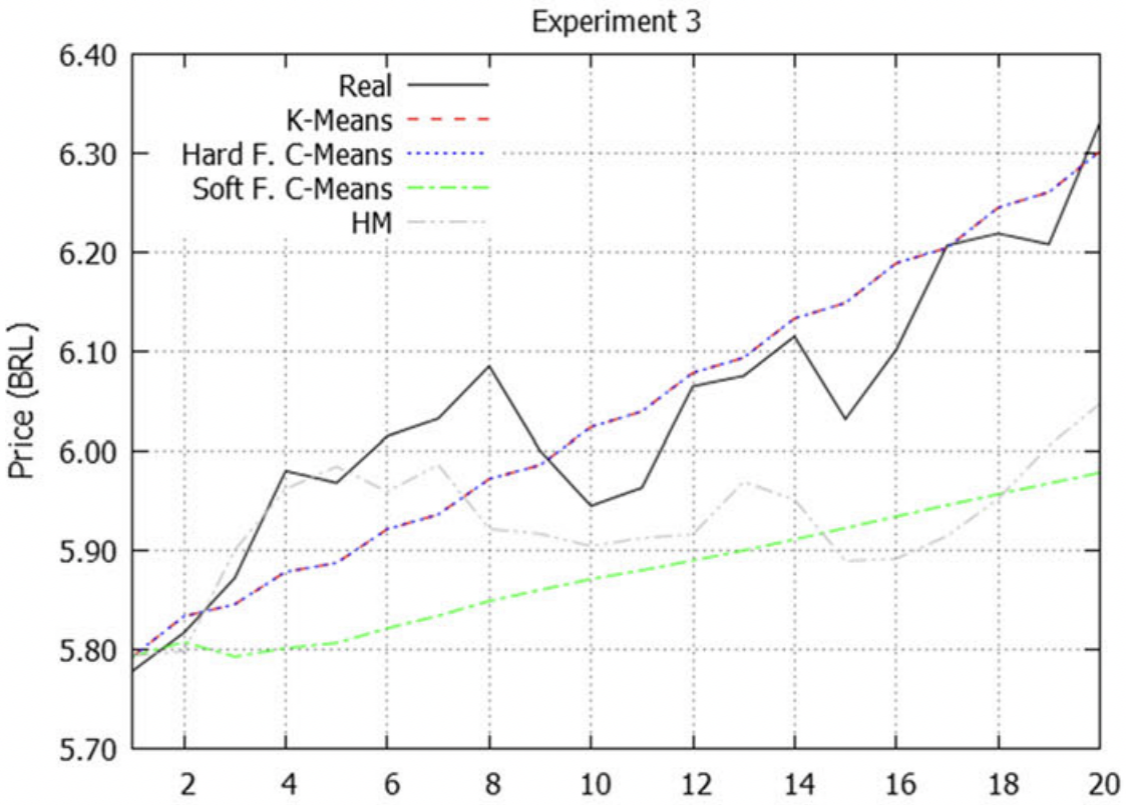
- Time Period: 01/07/2013 to 31/07/2013
- Context: This period is characterized by a clear uptrend movement in the stock price series.
- Volatility: Low

- Results: The proposed model using the K-Means method accurately follows the price trend and captures the price movement during this period. The accuracy of the model is better than that of the HM.



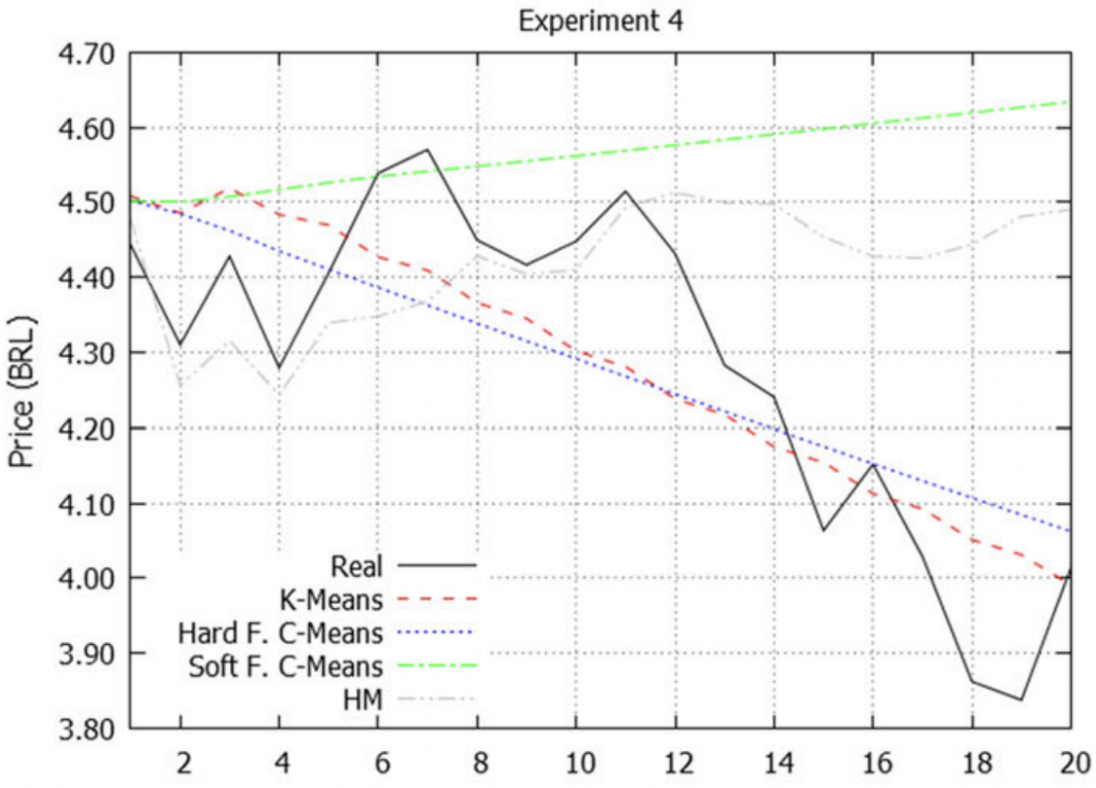
Experiment 2:

- Time Period: 01/08/2013 to 31/08/2013
- Context: This period continues the uptrend movement observed in Experiment 1.
- Volatility: Low
- Results: Similar to Experiment 1, the proposed model with the K-Means method successfully follows the price trend and captures the price movement. It outperforms the HM in terms of accuracy.



Experiment 3:

- Time Period: 01/09/2013 to 30/09/2013
- Context: This period is also part of the uptrend movement observed in Experiments 1 and 2.
- Volatility: Low
- Results: The proposed model with both the K-Means and Hard Fuzzy C-Means methods produces identical results because they generate the same clusters. The model accurately follows the price trend but with slightly lower intensity compared to the previous experiments. The accuracy of the proposed model remains better than that of the HM.



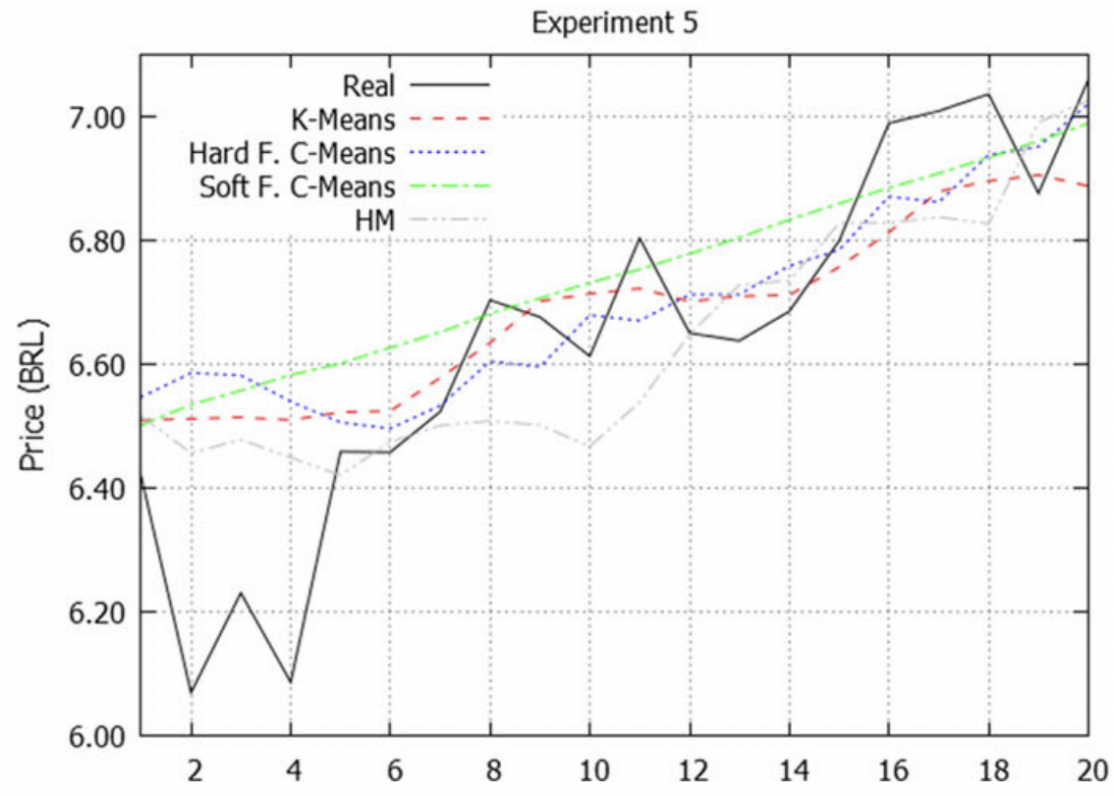
Experiment 4:

- Time Period: 01/10/2013 to 31/10/2013
- Context: This period represents a correction (downward) movement after a previous uptrend.
- Volatility: High
- Results: The proposed model with the hard-clustering methods, particularly the K-Means method, manages to capture the trend of the actual price series values during this period but with lower accuracy compared to the low volatility experiments. Other models, including the HM, perform even worse.

Experiment 5:

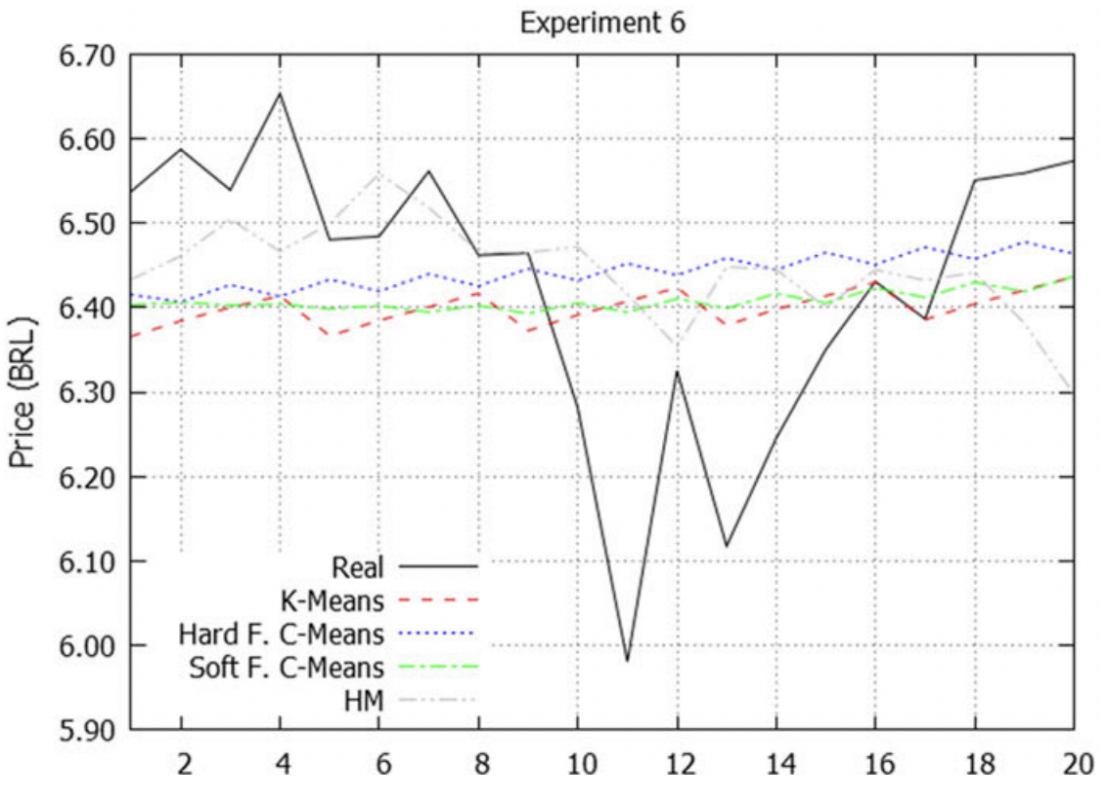
- Time Period: 01/11/2013 to 30/11/2013
- Context: This period corresponds to a horizontal movement occurring within an uptrend.
- Volatility: High

- Results: Similar to Experiment 4, the proposed model with the hard-clustering methods follows the trend of the actual price series values but with lower accuracy. Other models exhibit inferior performance.



Experiment 6:

- Time Period: 01/12/2013 to 31/12/2013
- Context: This period represents a long-term horizontal movement.
- Volatility: High
- Results: The proposed model with the hard-clustering methods shows mixed performance. While it manages to follow the trend of the actual price series values accurately in some instances, its overall accuracy is significantly lower compared to the low volatility experiments. Other models also perform poorly.



In summary, the experiments demonstrate that the proposed model using the hard-clustering methods performs better during periods of low volatility and clear price trends. However, its accuracy decreases in periods of high volatility. The HM and other models generally exhibit poorer performance across all experiments.

7 References

Reference Paper
https://link.springer.com/article/10.1007/s10462-018-9663-x

Book
https://cpb-us-w2.wpmucdn.com/blog.nus.edu.sg/dist/0/6796/files/2017/03/analysis-of-financial-time-series-copy-2ffgm3v.pdf

Code
https://colab.research.google.com/drive/16-3WYPJrc9M9kiEjKN6JeEFKN39Kl8JU#scrollTo=H9jD8M4pbH3o