kNN predictability analysis of stock and share closing prices

Thesis submitted for the degree of

Doctor of Philosophy

at the

University of Leicester

by

Yanshan Shi

Department of Mathematics

University of Leicester

Feburary 2016

"The important thing is not to stop questioning. Curiosity has its own reason for existing. One cannot help but be in awe when he contemplates the mysteries of eternity, of life, of the marvelous structure of reality. It is enough if one tries merely to comprehend a little of this mystery every day. Never lose a holy curiosity."

Albert Einstein

"A computer would deserve to be called intelligent if it could deceive a human into believing that it was human."

Alan Turing

Abstract

The k nearest neighbor rule or the kNN rule is a nonparametric algorithm that search for the k nearest neighbors of a query set in another set of points. In this thesis, application of the kNN rule in predictability analysis of stock and share returns is proposed. The first experiment tests the possibility of prediction for 'success' (or 'winner') components of four stock and shares market indices in a selected time period [1]. We have developed a method of labeling the component with either 'winner' or 'loser'. We analyze the existence of information on the winner–loser separation in the initial fragments of the daily closing prices log–returns time series. The Leave–One–Out Cross–Validation with the kNN algorithm is applied on the daily log–returns of components. Two distance measurements are used in our experiment, a correlation distance, and its proximity. By analyzing the error, for the HANGSENG and the DAX index, there are clear signs of possibility to evaluate the probability of long–term success. The correlation distance matrix histograms and 2–D/3–D elastic maps generated from the ViDaExpert show that the 'winner' components are closer to each other and 'winner'/'loser' components are separable on elastic maps for the HANGSENG and the DAX index while for the negative possibility indices, there is no sign of separation.

In the second experiment, for a selected time interval, daily log–return time series is split into "history", "present" and "future" parts. The kNN rule is used to search for nearest neighbors of "present" from a set. This set is created by using the sliding window strategy. The nearest neighbors are considered as the predicted "future" part. We then use ideas from dynamical systems and to regenerate "future" part closing prices from nearest neighbors log–returns. Different sub–experiments are created in terms of the difference in generation of "history" part, different market indices, and different distance measurements. This approach of modeling or forecasting works for both the ergodic dynamic systems and the random processes. The Lorenz attractor with noise is used to generate data and the data are used in the kNN experiment with the Euclidean distance. The sliding window strategy is applied in both test and training set. The kNN rule is used to find the k nearest neighbors and the next 'window' is used as the prediction. The error analysis

of the relative mean squared error RMSE shows that $k = 1$ can give the best prediction and when $k \to 100$, the average RMSE values converge. The average standard deviation values converge when $k \to 100$. The solution $Z(t)$ is predicted quite accurate using the kNN experiment.

Acknowledgements

It is my pleasure to acknowledge the help and support I have received from many people during a four and half years Ph.D life at University of Leicester. I would like to take this opportunity to acknowledge a few.

Firstly, I would like to express my utmost gratitude to my supervisor, Prof. Alexander N. Gorban, for his support, time and patience, who also been my inspiration throughout my studies. I must say thanks to him for all of the help, encouragement and suggestions which are by no means limited to my research but have also proven valuable in my life. This thesis draws on his talents, knowledge and contribution.

I would like to thank all of staff in College House for their help and making so many things easier. I am grateful to Dr. Ivan Tyukin, Prof. Jeremy Levesley, Dr. Evgeny Mirkes and Prof. Sergei Petrovskii.

On a personal level I would thank all of my colleagues from Maths department, Ayo, Ruhao, Zexun, Wenyan etc. Special thanks go to Juxi, Jianxia and Masha who provided all the ultimate care, support and enjoyable time.

Finally, it gives me the most pleasure to thank my parents for the love, faith and support. I would like to thank my girlfriend Beiyao, for her love and support. Without their help this thesis would not be possible.

Contents

List of Figures

List of Tables

To my parents

Chapter 1

Introduction

Concepts of data mining have been developed for decades and it has become a popular subject in today's world. This subject is a combination of computer science and statistics, and it is widely applied in different areas such as decision making, prediction and forecasting, pattern recognition, artificial intelligence and so on. Data mining techniques have been developed and applied in the prediction of stock and share returns and predictability studies. The future stock returns can be well forecast by applying neural network models and cross–validation is applied to improve the generalization ability of selected models [2]. A study of finding non–linear regularities of asset price movements by using the neural network and learning methods would be another application on applying data mining techniques on IBM daily stock returns [3]. A self–organizing fuzzy neural network is computed by using historical data of Dow Jones Industry Average and selected twitter mood dimensions can improve the prediction accuracy of DJIA closing prices [4]. In prediction process of five selected components of Jordanian market, by setting $k = 5$ and prepared training dataset, the k Nearest Neighbor algorithm is applied and the predicted results are close to the real prices [5]. Combinations of several data mining algorithms such as k Nearest Neighbor, neural network, and decision tree are tried to predict the DJVA closing prices. After applying the Hurst exponent, the prediction accuracy for chosen period is higher [6]. k Nearest Neighbor algorithm is simple and it has no models to fit. The generalizations of this algorithm are reviewed in the next section. In this thesis, it is used in predictability analysis for stock and share closing prices time series and modeling the stock and share closing prices time series.

1.1 The reviews of (k) Nearest Neighbor rules

The nearest neighbor technique is widely applied in many fields such as pattern recognition, text categorization, object recognition. This technique is generalized and surveyed in [7]. Many NN techniques are reviewed and they are classified into two major types, the structure–less techniques and structure–based techniques. Different methods are described in terms of memory limitation and computational complexity. In general, the structure–less techniques required no model to fit.

1.1.1 The structure–less Nearest Neighbor rules

k Nearest Neighbor (or kNN) is one of the simplest algorithms of this type. This algorithm is able to classify a test point based on voting of the nearest k neighbors (i.e. when nearest neighbors are searched, the label of test point depends on the majority votes of k nearest neighbor labels). The term 'nearest' can be measured in terms of distance function of various forms (i.e. Euclidean distance). This algorithm has a fast training time and is easy to implement. However, a large memory is required. Also, the choice of k becomes a problem since it depends on the structure of the data. Another drawback is that this algorithm can be easily affected by irrelevant attributes. The classification results kNN rule are affected by the structure of dataset and the choice of the parameter k [8].

The first k Nearest Neighbor discriminatory rule was introduced in [9]. Let $X_1, X_2, \ldots, X_m$ be observations from an unknown distribution F and $Y_1, Y_2, \ldots, Y_n$ be observations from unknown distribution G. Suppose we have an observation Z from an unknown distribution which is neither F nor G. All observations are defined in p–dimensional space. By defining the concept of 'closeness' in p–dimensional space between Z and all observations from distributions F and G, the discriminatory rule is then defined as assigning Z to distribution F if the majority of its k nearest observations are from distribution F and assigning Z to distribution G otherwise, for a chosen odd number k. Also, if F and G are known, then as $m \to \infty$ and $n \to \infty$, we have:

$$\begin{aligned} P_1 &= Pr(Z \text{ is assigned to } F | Z \text{ comes from } F), \\ P_2 &= Pr(Z \text{ is assigned to } G | Z \text{ comes from } G), \end{aligned} \tag{1.1}$$

such that P_1 and P_2 will achieve theoretical minimum values. It is noted that no restrictive assumptions on F, G or measure of distance.

After Fix and Hodge discussed the kNN discriminantory rule, error analysis and study of admissibility for the rule is discussed. For a classification problem in data mining, the nearest neighbor rule simply classifies the sample point with unknown label with the label of nearest set of points. It is shown that for the nearest neighbor rule (1NN rule) in the n–sample problem, the probability error is:

$$P_e(1;n) = (\frac{1}{2})^n,$$

and if we set $k = 2k_0 + 1$, the probability error for the kNN rule in this problem is:

$$P_e(k;n) = \left(\frac{1}{2}\right)^n \sum_{j=0}^{k_0} \binom{n}{j}.$$

Since $P_e(1;n) < P_e(k;n)$ with $k \neq 1$, the 1NN rule is admissible among kNN rule for n–sample problems [10].

Consider (x, θ) and (x_1, θ_1), (x_2, θ_2), $\ldots$, (x_n, θ_n) be $n+1$ idenpendently identically distributed random variables with values in $X \times \Theta$ where X in metric space ρ and Θ in an abstract parameter space. Nearest neighbor rule estimate θ of x to be θ'_n with known (x_1, θ_1), (x_2, θ_2), $\ldots$, (x_n, θ_n). The loss function L is defined on $\Theta \times \Theta$ assigns loss $L(\theta, \hat{\theta})$ such that $\hat{\theta}$ estimates the true parameter θ. The conditional Bayes risk $r^*(x)$ and unconditional Bayes risk R^* are defined as:

$$\begin{aligned} r^*(x) &= \underset{\theta}{E}\{L(\theta, \theta^*(x)|x\} \leq \underset{\theta}{E}\{L(\theta, \hat{\theta})|x\}, \\ R^* &= \underset{\theta}{E} r^*(x), \end{aligned} \tag{1.2}$$

where:

$$\underset{\theta}{E}\{L(\theta, \theta^*(x))|x\} = \int_\Theta L(\theta, \theta^*(x)) f(\theta|x) \mathrm{d}\theta, \tag{1.3}$$

and

$$\underset{\theta}{E} r^*(x) = \int_X r^*(x) f(x) \mathrm{d}x = \int_\Theta \int_X L(\theta, \theta^*(x)) f(\theta, x) \mathrm{d}x \mathrm{d}x, \tag{1.4}$$

such that the probability densities $f(x, \theta)$, $f(\theta|x)$, $f(x)$ exist.

Let us define $x'_n \in \{x_1, x_2, \ldots, x_n\}$ the nearest neighbor to x and θ'_n is the associated parameter. The conditional n –sample NN risks are defined as:

$$r_n(x, x'_n) = \underset{\theta, \theta'_n}{E}\{L(\theta, \theta'_n)|x, x'_n\},$$

and

$$r_n(x) = \underset{x'_n}{E} r_n(x, x'_n).$$

The asympotic conditional NN risk is defined as:

$$r(x) = \lim_{n\to\infty} r_n(x).$$

The unconditional n–sample NN risk R_n is defined as:

$$R_n = \underset{\theta}{E} r_n(x) = \underset{\theta,\theta'_n}{E} L(\theta, \theta'_n),$$

and when n is large, the NN risk R is defined as:

$$R = \lim_{n\to\infty} R_n = \lim_{n\to\infty} \underset{\theta,\theta'_n}{E} L(\theta, \theta'_n).$$

An important proof from Cover and Hart's paper is that by applying Bayes procedure, the probability error R of the NN rule in the classification problem with M categories is bounded as:

$$R^* \leq R \leq R^*(2 - \frac{MR^*}{M-1}),$$

such that these boundaries are the tightest possible where R^* is Bayes error [10].

The rate of convergence of NN risk R_n is investigated. It is concluded that the n–sample NN risk R_n converges to R on the order of $1/n^2$ [11]. The NN classifier result in $(n+1)$–sample case is different from the result in n samples only if the $(n+1)$ sample is the closest and the probability when this happens is $1/(n+1)$. And we have:

Theorem 1.1. *$|R_n - R_{n+1}| \leq 1/(n+1)$, and in general $R_n - R_{n+k} \leq k/(n+1)$.*

The theorem below shows that the rate of convergence for R_n:

Theorem 1.2. *Let $f_1(x)$ and $f_2(x)$ have unifromly bounded thrid derivatives and be bounded away from zero on their (probability one) support sets. Then $R_n = R_\infty + \mathrm{O}(1/n^2)$.*

For large sample size, Under mild continuity conditions, the conditional risk of NN rule satisfies:

$$\begin{aligned} &r(x) \leq 2r^*(x), \text{formetricloss}, \\ &r(x) = 2r^*(x), \text{forsquared} - -\text{errorloss}. \end{aligned} \tag{1.5}$$

The unconditional NN risk R under certain additional moment conditions satisfies:

$$\begin{aligned} &R \leq 2R^*, \text{formetricloss}, \\ &R = 2R^*, \text{forsquared} - -\text{errorloss}, \\ &R = (1 + 1/k)R^*, \text{forsquared} - -\text{errorlosswitha}k\text{NNestimate}. \end{aligned} \tag{1.6}$$

These results show that when n is large, the NN risk no greater than twice the Bayes risk for both the squred–error loss function and the metric function [12].

After the kNN algorithm formulated, development of this algorithm could classify the unclassified samples with faster computational speed [13]. A preprocessing method that could reduce the computational time for classifying an unknown sample is developed. This method is aimed to partition the featuring space in order to speed up the kNN algorithm. Then the point is classified use information from the centroid of these feature space and the idea is to compute fewer distances than the original kNN algorithm. The new algorithm saved some storage but enlarge the computational complexity since the computation of partitioned feature spaces can be complicated.

One of the drawbacks of the nearest neighbors rule is caused by the storage of distances. To find the nearest neighbor for a test point, all distances between the test point and every point in the training set are computed. The Nearest Neighbor rule (NN rule) for a given point search for its nearest neighbor from this set of points. The Condensed Nearest Neighbor rule (or CNN rule) select the subset of this set of points and then apply NN rule for a given test point [14]. It is assumed that the data is in some orders before applying this algorithm. Let us define STORE and GRABBAG to be two bins. The algorithm for the CNN rule is listed in Algorithm 1. The input is the training set and output is the subset of this set contains prototype points selected using CNN rule.

CNN rule select data points as 'prototype points' based on Bayes risk. A small Bayes risk leads to a small overlap of underlying densities of the different classes. In this case, CNN rule picks out the points near the boundary between classes. For a large Bayes risk, all data points will be put in STORE hence there will be no

Algorithm 1 Condensed Nearest Neighbor (CNN) rule

1. Place the first data point of an input set of data in STORE.
2. Use NN rule to classify the second data point is classified with training set is set to be current STORE. Put the second data point in GRABBAG if it is correctly classified. If not, put this data point in STORE.
3. Repeat the process such that: put the ith data point in GRABBAG if it is correctly classified with the training set of current STORE. If not, put the ith data point in STORE.
4. After applying this procedure once for every point in the input set of points, it continues to run through GRABBAG. This procedure will be terminated if one of the two conditions is satisfied:
   (a) There is no data point in GRABBAG. i.e. all data points are now put into STORE.
   (b) After applying this procedure once for every point in GRABBAG and there are no more points transferred into STORE.
5. The set of data points in STORE when the algorithm finishes are used as the training set for NN rule and data points from GRABBAG are abandoned.

Algorithm 2 Fast Condensed Nearest Neighbor (FCNN) rule

$S = \emptyset$;
$\Delta S = Centroids(T)$;
while $\Delta S \neq \emptyset$ **do**
  $S = S \cup \Delta S$;
  $\Delta S = \emptyset$;
  for each $(p \in S)$ **do**
    $\Delta S = \Delta S \cup rep(p, Vor(p, S, T))$
  end for
end while
  return S

important reduction. This rule is introduced to solve the storage problem of NN rule by reducing the size of the training set. Fast Condensed Nearest Neighbor rule (FCNN rule) is studied based on the idea of CNN. This rule aims to compute a consistent subset of the training set with order independent and sub–quadratic worst case time complexity but the convergence rate to the final subset of the training set is faster [15]. Consider a training set T with labels and the distance metric is defined as d. The label of p is defined as $l(p)$. For a test point q, NN rule assign label of q with the label of the nearest neighbor of q from the training set T. Let S be a consistent subset of T such that we have for every $p \in T$, we have $l(p) = NN(p, S)$. Let $\mathrm{Vor}(p, S, T)$ be the set $\{q \in T | \forall p' \in S, \mathrm{d}(p, q) \leq \mathrm{d}(p', q)\}$ and let $\mathrm{Voren}(p, S, T)$ be the set $q \in \mathrm{Vor}(p, S, T) | l(q) \neq l(p)$. Let $Centroids(T)$ be the set of centroids of each class label in T. The algorithm for fast condensed nearest neighbor rule is defined in Algorithm 2.

From the result of the experiment, FCNN rule can have a relatively competitive accuracy in terms of the size of the model. The speed of convergence to the resultant subset of original training set is generally faster than the original CNN rule.

The classification problem such as assignment of a given input to one of several classes is widely studied. The Condensed Nearest Neighbor rule (CNN) before using the formal algorithm of solving classification problem. Let d_{ji} be the result computed from the learning method when the classifier is j for class i. The voting is to assign the input to the class c with maximum vote:

$$r_i = \sum_{j=1}^{m} d_{ji}\beta_j,$$

$$x = arg \max_{i=1}^{n}[r_i].$$

The weights $\beta_j \geq 0, \sum_{j=1}^{m} \beta_j = 1$. In simple voting, the weights are taken as the same: $\beta_j = 1/m$. For weighted voting, β_j is computed as: if for classifier j, e is the most probable class and f is the next most probable,

$$\alpha_j = p_j(e|x) - p_j(f|x),$$

and $\beta_j = \alpha_j / \sum_{l=1}^{m} \alpha_l$. These two types of voting techniques are reviewed and compared. It is concluded that when there is a relatively small training set, voting could improve the results for specific problems; when there is a relatively large training set, the voting results converge [16].

An updated version of CNN rule is introduced with concept of mutual nearest neighborhood and a new measurement of similarity named the mutual neighborhood value or (MNV for short) [17]. Consider a set of data points $X_1, X_2, \ldots, X_N$ in L–dimensional metric space with metric d. Let X_j be the mth nearest neighbor of X_i and let X_i be the nth nearest neighbor of X_j. The mutual neighborhood value is defined as:

$$\mathrm{MNV}(X_i, X_j) = \begin{cases} 0 & \text{if } i = j \\ m + n & \text{otherwise,} \end{cases} \tag{1.7}$$

where $m \in \{0, 1, 2, \ldots, N-1\}$ and $n \in \{0, 1, 2, \ldots, N-1\}$. The modified algorithm for CNN rule using concept of MNV is listed in Algorithm 3. There are

Algorithm 3 Two–stages algorithm of CNN rule using the concept of MNV

Stage 1

1. For each data point X of the training set T, apply NN rule to find the nearest neighbor Y among data points from opposite class. Then for Y, find the nearest neighbor rank for X from the opposite class, i.e. rank J of X. The MNV of X with respect to Y is $\text{MNV}(X, Y) = 1 + J$. This MNV is associated with X alone and not with Y. The Euclidean distance d between X and Y is computed and associate this value with X. Hence, data points that are near the decision boundary will have low values of MNV and d.
2. After computing MNV for all data points, sort all data points by MNV in ascending order. If some of the MNV's are equal, these data points can be sorted by d in ascending order. Put the ordered data points in ORDER.
3. Put the first data point of ORDER in STORE.
4. The next data point in ORDER is classified by NN rule uses points in STORE as the training set. Compare the original label with the classified label, if they are not the same, this data point is misclassified .If this data point is misclassified, put it in STORE.
5. Step 4 is repeated until every data point in ORDER is classified.
6. After running this process through every point in ORDER once, apply last step 4 and step 5 to the points left in ORDER. This procedure is terminated when there are no more transfers of data points from ORDER to STORE.

Stage 2

Further reduction is then required to make a modified condensed set.

7. Put a data point X from STORE (result from step 6 in SIFT.
8. Apply NN rule on every data point in ORDER and they are classified using the current STORE as the training set. If it is misclassified, put Z back to STORE; if it is successfully classified, keep it in SIFT.
9. Repeat step 7 and step 8 for all data points in STORE.

two stages for this algorithm and input is a set T of data points and the output is the subset of T.

The STORE, in the end, is the output of this algorithm. By comparing the results of modified CNN rule with original CNN rule, the modified rule is able to deal with the case if a data point is not near the decision boundary. However, it requires more steps to get the condensed set of data points.

An extension of the CNN rule is studied to have a further reduction on the data set. Reduced Nearest Neighbor rule (or RNN rule) is introduced and the comparison is studied with original CNN rule [18]. Consider there are M classes for a data set. Let each data point is defined in N–dimensional feature space and for each class, there are K data points in the training set. Each data point is associated with a class. Each pair of them is defined as: (x^i, θ_i), where $1 \leq i \leq K$ and $\theta_i \in \{1, 2, \ldots, M\}$. Let $x^i = (x_1^i, x_2^i, \ldots, x_N^i)$ denote the set of

Algorithm 4 Algorithm of Reduced Nearest Neighbor Rule (RNN rule)

1. Put all data points of T_{CNN} in T_{RNN}.
2. Remove the first data point in T_{RNN}.
3. Classify all data points of T_{NN} by using T_{RNN} as training set:
   (a) If all data points are correctly classified, then proceed to the step 4.
   (b) If one of data points is misclassified, then put the removed data point back to T_{RNN} and go to step 4.
4. If all data points in T_{RNN} are removed once or replaced once, the algorithm terminates. If not, remove the next data point in T_{RNN} go to step 3.

feature values for data point x^i. The nearest neighbor training set is defined as $T_{\text{NN}} = \{(x^1, \theta_1), (x^2, \theta_2), \ldots, (x^K, \theta_K)\}$. Let T_{CNN} be the result set of data points from original CNN rule. The algorithm of RNN rule with input of set T_{CNN} and output is a set T_{RNN} is described in Algorithm 4.

The idea of RNN rule is to reduce the size of the training set by deleting the points that are not affecting the classification results. From the experiment on IRIS dataset, it shows that original CNN rule has good performance while RNN rule is slightly better. However, the complexity of cost in computation of the algorithm is higher.

The "curse of dimensionality" has been studied for Nearest Neighbor rule (NN rule) and there is evidence show that in high dimensional space, the results of NN rule has bias [19] [20]. Discriminant Adaptive Nearest Neighbor rule (or DANN rule) is studied to try to reduce this bias by estimating a new metric for searching nearest neighbors [21]. This estimation idea could be concluded as searching for local boundaries with details given in centroid then use these local boundaries to shrink the original neighborhood in directions which are orthogonal to the local boundaries. The summary of the iterating procedure is described in Algorithm 5.

Hence, this adaptive method of DANN rule has increased the performance of original NN rule for some problems.

kNN rule has been considered as one of important data mining algorithms by its applications in many areas. kNN rule is used to detect intrusion of program behaviour and it is proven to be effective when 1998 DARPA BSM audit data is used in experiment that kNN rule can detect intrusive attacks [22]. kNN rule is applied to in text categorization by using training documents [23] [24]. This rule is also applied tin diagnosis of breast cancer [25], pattern classification [26], face detection [27], image classification [28], regression problems [29], query dependent

Algorithm 5 Iterating procedure for Discriminant Adaptive Nearest Neighbor rule

1. The metric Σ is initialized as identity matrix $\Sigma = \mathbf{I}$.
2. For a test point $\mathbf{x}_0$ the nearest neighborhood of K_M points are launched in metric Σ, where K_M is the number of nearest neighbors in the neighborhood for estimation of metric.
3. For points in the neighborhood, use them to compute the weight within sum–of–square matrix $\mathbf{W}$ and the weighted between between sum–of–square matrix $\mathbf{B}$.
4. Update the metric $\Sigma = \mathbf{W}^{-1/2}(\mathbf{W}^{-1/2}\mathbf{B}\mathbf{W}^{-1/2} + \epsilon)\mathbf{W}^{-1/2}$, where ϵ is a parameter in the metric.
5. Repeat steps 2 3 4.
6. When iteration process stops, metric Σ is used for kNN classification at test point $\mathbf{x}_0$.

ranking [30], multivariate time series analysis [31] [32], time series classification [33] etc.

1.1.2 The structure–based Nearest Neighbor rules

The second type of the technique requires the structure of data before applying the algorithms. Examples of this typed techniques are ball tree nearest neighbor (or KNS1), k–d tree nearest neighbor (or kdNN), principal axis tree (or PAT), orthogonal structure tree (or OST), nearest feature line (or NFL) etc. Ball tree structure is used such that the nearest neighbor is searched efficiently from internal nodes and their information is stored in the leaves of the tree. This method has good performance for high dimensional entities and the implementation is easy [34]. However, the drawback is it is expensive to use insertion algorithm and the increase of distance would make KNS1 degrade. The Feature Line (FL) can be used in the computation of nearest neighbor. For each class, the distance between each point in the test set and each pair of FL is computed. Then the set of distances are sorted into ascending order and the Nearest FL distance is assigned as rank 1. This algorithm has an improved accuracy in classification problem and it is efficient for a smaller dataset. It also uses information that is ignored in the nearest neighbor. The drawback of this rule is that this fails when the prototype in NFL is far away from the test point. The computational complexity is higher. It is also difficult to use a straight line to represent the data points. The local nearest neighbor rule which evaluates feature line and the feature point in each class can improve the performance of NFL rule. Instead of concentrating on data points, this algorithm

focuses on the nearest neighbor prototype of the test points. Hence, it covers the drawbacks of NFL rule. However, due to the modified method, the number of computations is higher. In NFL rule, the training set is transformed into feature line. Then distances between test points and feature lines are computed where the nearest neighbor is selected with the smallest distance between feature line and test point. This method could have improvement on accuracy of classification results and it is efficient for a relatively small size of dataset [35]. A local nearest neighbor classifier is studied to improve the performance of NFL rule. This method focuses on computing the feature points and feature lines and uses the points only as prototypes, search for the nearest neighbor of the test points [36]. PAT rule transforms the training set into more efficient form for searching of nearest neighbors [37]. PAT is constructed first by using principal component analysis and then splits the whole dataset into a number of a subset with the same size. Then kNN rule is used to search for nearest neighbor in this tree. The speed of searching for nearest neighbors is fast but the process is becoming more complex. An update of PAT rule is OST rule. OST rule uses the idea of orthogonal vector and the search tree is constructed with a root node. Then all data points are assigned to this node [38]. Comparing with PAT rule, it shorten the computation time for tree construction and it is efficient for large dataset. In this paper, methods of structure–less are mainly studied and applied.

1.2 Some backgrounds of research

In this section, we recall the basic mathematical notions and present the background of the method developed in the thesis. The predictability of log–return time series is analyzed by using different experiments constructed from ideas of kNN in this thesis. Time series is the main object of research. It has important properties such as stationarity and ergodicity. A time series can be discrete or continues. Discretization is considered as an important way to transfer a continues time series to discrete state. kNN algorithm has convergence properties for density. The combination of this properties and the convergence in conditional probability is applied to show that our experiment works for both ergodic dynamic systems and ergodic random process. This would again imply further to the assumptions of technical analysis and efficient market hypothesis. In this section, some backgrounds are introduced.

1.2.1 Time series, random processes, stationarity, and ergodicity

A time series is a sequence of numbers over a time interval. It is widely seen in present life, especially in financial and econometric disciplines. An example of time series could contain prices of stock and shares, prices of financial securities, daily temperature, earthquake indicator etc. Properties of time series is always an interesting question for researchers. Time series can be defined in both continuous or discrete frameworks. A discrete time series has a value at each specific time and this value is unchanged at this time. A continues time series has a value for a very short amount of time and in other words, for a value, there are infinite numbers of time data between two time data. Discretization is a method applied to a continuous function or data and transfers them into discrete parts. An application of discretization of time series is the clustering discretization of time series, the time series is used to discover rules and patterns [39]. In analysis of predictability in this thesis, sliding window strategy is applied as discretization of input time series.

From probabilistic point of view, we consider a time series a trajectory of a random process in discrete time.

Definition 1.3. A random process in discrete time is a random variable X_t which depends on time $t \in Z$. A random process is given if for every finite set of time moments $\{t_1, \ldots, t_n\} \subset Z$ and for every Borel set U in R^n a probability of a cylindrical set $P((X_{t_1}, X_{t_2}, \ldots, X_{t_n}) \in U)$ is defined.

Extension of the probability distribution on the sigma–algebra of sets produced from the cylindric sets by countable unions, intersections and subtractions should satisfy the probability axioms.

Remark 1.4. It is sufficient to define the probability on the simplest cylindrical sets with sufficiently small cubic backgrounds

$$U = \{(X_{t_1}, X_{t_2}, \ldots, X_{t_n}) \mid |X_{t_i} - x_i| < \varepsilon\},$$

for any set of numbers $\{x_1, \ldots, x_n\}$ and $\varepsilon > 0$

For the mathematical theory of random processes we refer to the classical books [40] [41]. Examples of random process include simple random walk, Brownian motion etc.

Definition 1.5. Consider a sequence of independent and identically distributed random variables $(X_n; n \geq 1)$. Set

$$S_n = \sum_{r=1}^{n} X_r + S_0$$

Then $(S_n, n \geq 0)$ is a general random walk. Consider a sequence of independent random variables $(X_n; n \geq 1)$ such that:

$$P(X_n = 1) = p, P(X_n = -1) = q, p + q = 1, n \geq 1.$$

Then $S_n = S_0 + \sum_{r=1}^{n} X_r$ is called simple random walk on integers starting from some integer S_0. This random walk is symmetric if $p = q = \frac{1}{2}$ [42].

Random processes can be considered as collections of time series trajectories, for example, Brownian motion, random walks [43] etc. Brownian motion is defined as below.

Definition 1.6. A Brownian motion is a stochastic process $\{X(t); t \geq 0\}$ with following properties [44]:

1. Every increment $X(t + s) - X(s)$ is normally distributed with mean 0 and variance $\sigma^2 t$; σ is a fixed parameter.

2. For every pair of disjoint time intervals $[t_1, t_2]$, $[t_3, t_4]$, say $t_1 < t_2 \leq t_3 < t_4$, the increments $X(t_4) - X(t_3)$ and $X(t_2) - X(t_1)$ are independent random variables with distributions given in 1, and similarly for n disjoint time intervals where n is an arbitrary positive integer.

3. $X(0) = 0$ and $X(t)$ is continuous at $t = 0$.

One of most famous studies of financial time series is prediction or forecast of future prices when the historical prices are given. Every financial time series exists in a unique version and cannot be 'restarted'. Therefore, for empirical evaluation of the probabilities (through frequencies) we have to consider different time series as trajectories of the same ('universal') random process (may be, after some transformations) or to use hypotheses about stationarity (stationary property of time series) and ergodicity (ergodicity property of time series). For example,

researchers use conditions of time series such as stationarity and ergodicity to construct methods of prediction.

Definition 1.7. A random process X_t $(t \in Z)$ is stationary if for every finite set of time moments $\{t_1, \ldots, t_n \subset Z\}$, for every Borel set U in R^n, and every time shift $\delta \in Z$

$$P((X_{t_1+\delta}, X_{t_2+\delta}, \ldots, X_{t_n+\delta}) \in U) = P((X_{t_1}, X_{t_2}, \ldots, X_{t_n}) \in U).$$

This means that the probability does not depend on the shift of time scale. White noise is an example of a stationary random process. White noise is widely studied in area of signal processing, telecommunications, statistical forecasting, etc. The characteristics of white noise is studied using the empirical mode decomposition method (or EMD method) [45].

Definition 1.8. A process X_t is white noise if its values X_{t_i} and X_{t_j} are uncorrelated for every t, and $t_j \neq t_i$:

$$\rho(t_i, t_j) = 0,$$

for $t_i \neq t_j$. It is assumed that the mean of a white noise process is identically 0 [46].

Definition 1.9. A stationary random process X_t $(t \in Z)$ is ergodic if the expectation of every measurable function $g(\mathbf{X})$ $(\mathbf{X} = (\ldots, X_{-t}, X_{-t+1}, \ldots, X_{t-1}, X_t, \ldots))$ can be computed by time average on a trajectory $\mathbf{x} = (\ldots, x_{-t}, x_{-t+1}, \ldots, x_{t-1}, x_t, \ldots)$: with probability one [47] [48]:

$$E[g(\mathbf{X})] = lim_{T\to\infty} \frac{1}{T} \sum_{\delta=0}^{T-1} g(\mathbf{x}_\delta),$$

where the sequence $\mathbf{x}_\delta$ is produced from $\mathbf{x} = \mathbf{x}_0$ by the time shift on δ: in $\mathbf{x}_\delta$ the values of variables X_t are: $X_t = x_{t-\delta}$.

The following proposition is just a law of large numbers applied to a stationary ergodic random process [49].

Proposition 1.10. *Let X_t $(t \in Z)$ be a stationary ergodic random process, x_t $(t \in Z)$ be a trajectory of this process, $\{t_1, \ldots, t_n \subset Z\}$, $U \subset R^n$ be a Borel set*

and

$$h_U(\tau) = \begin{cases} 1 \text{ if } (x_{t_1+\tau}, x_{t_2+\tau}, \ldots, x_{t_n+\tau}) \in U, \\ 0 \text{ if } (x_{t_1+\tau}, x_{t_2+\tau}, \ldots, x_{t_n+\tau}) \notin U. \end{cases}$$

Then

$$\frac{1}{T}\sum_{\tau=0}^{T-1} h_U(\tau) \to P((X_{t_1}, X_{t_2}, \ldots, X_{t_n}) \in U), \tag{1.8}$$

with probability one when $T \to \infty$ ($T \in Z$).

Remark 1.11. In Definition 1.9 and Proposition 1.10 we can use averaging on $[0, T]$, $[0, -T]$, or symmetric intervals $[-T, T]$ ($T \to \infty$). These methods of averaging will give the same results because of the stationarity assumption.

There are many tests for detection of non–stationarity of time series and many tools for the transformation of time series to a stationary form. Evidence has shown that macroeconomic time series are well characterized if they are transformed into stationary time series with a trend [50]. For example, a random walk is obviously non–stationary because its variance grows in time but the time series of returns, $X_{t+1} - X_t$ or log–returns $\ln(X_{t+1}/X_t)$ (for a geometric random walk) may be stationary. Other standard tools are detrending and extraction of seasonality. Econometric time series can be detrended into a random walk with a drift and the detrended data could be used in a regression model [51]. For climate data, an adaptive Empirical Mode Decomposition (EMD) method is used as a detrending algorithm [52].

Filters in time series transform an input sequence of data to a required sequence of data for different purposes. Band–pass filter is a type of filters that delete frequencies of a pre–defined range. Low–pass filters are filters which aim to extract and delete the low–frequency components from sequences of data. For example, an RLC filter is applied in an RLC circus to change the frequency of input data [53] [54]. In application of time series, infinite impulse response and impulse response filters use Fourier transformations are normally used.

1.2.2 The prediction problem and the conditional probability

The problem of prediction is of great practical importance. Suppose we know a piece of history: $X_0 = x_0, X_{-1} = x_{-1}, \ldots, X_{-k} = x_{-k}$. We need to evaluate the future development of the process, i.e., we need the probability distribution for

$(X_1, X_2, \ldots, X_m)$ for some $m > 0$. This problem may be formulated as estimation of conditional probabilities:

$$P((X_1, \ldots, X_M) \in U | X_0 = x_0, X_{-1} = x_{-1}, \ldots, X_{-k} = x_{-k}),$$

for an *open set* U with non-zero probabilistic measure. Let B_ε^n ($\varepsilon > 0$) be a ball in R^n of radius ε with center at the origin. Assume the U is an open set. Almost everywhere

$$\begin{aligned} &P((X_1, \ldots, X_m) \in U | X_0 = x_0, X_{-1} = x_{-1}, \ldots, X_{-k} = x_{-k}) \\ &= \lim_{\varepsilon \to 0} P((X_1, \ldots, X_M) \in U | (X_0, X_{-1}, \ldots, X_{-k}) \in (x_0, x_{-1}, \ldots, x_{-k}) + B_\varepsilon^{k+1}). \end{aligned} \tag{1.9}$$

This use of the vicinity of the history $((X_0, X_{-1}, \ldots, X_{-k}) \in (x_0, x_{-1}, \ldots, x_{-k}) + B_\varepsilon^{k+1})$ instead of exact values is necessary for estimation of probabilities from statistics of empirical samples. At the same time, from the practical point of view, we never have an absolutely precise real values of x, therefore use of the vicinities is even more reasonable than the exact values.

Combine (1.9) with (1.8) the estimate of conditional probability is produced. Let X_t ($t \in Z$) be a stationary ergodic random process, x_t ($t \in Z$) be a trajectory of this process. For a given piece of history $X_0 = x_0, X_{-1} = x_{-1}, \ldots, X_{-k} = x_{-k}$, $m \in Z_+$ (the depth of prediction and $\varepsilon > 0$ find all $\tau < -m$, for which

$$(x_\tau, x_{\tau-1}, \ldots, x_{\tau-k}) \in (x_0, x_{-1}, \ldots, x_{-k}) + B_\varepsilon^{k+1}.$$

Denote this set $\mathbf{T}_\varepsilon$. For each τ 'the future of length m' is defined as $(x_{\tau+1}, x_{\tau+2}, \ldots, x_{\tau+m})$.

Proposition 1.12. *For $\tau \in Z$ define*

$$\theta_{\varepsilon,U}(\tau) = \begin{cases} 1 \text{ if } (x_{\tau+1}, x_{\tau+2}, \ldots, x_{\tau+m}) \in U, \\ 0 \text{ if } (x_{\tau+1}, x_{\tau+2}, \ldots, x_{\tau+m}) \notin U. \end{cases}$$

Then with probability one

$$\begin{aligned} &P((X_1, \ldots, X_m) \in U | (X_0, X_{-1}, \ldots, X_{-k}) \in (x_0, x_{-1}, \ldots, x_{-k}) + B_\varepsilon^{k+1}) \\ &\qquad = \lim_{T \to \infty} \frac{1}{|\mathbf{T}_\varepsilon \cap [0, -T]|} \sum_{\tau \in \mathbf{T}_\varepsilon, \tau > -T} \theta_{\varepsilon,U}(\tau), \end{aligned} \tag{1.10}$$

and for conditional probability we get (almost always, with probability one).

$$P((X_1,\dots,X_m)\in U|(X_0,X_{-1},\dots,X_{-k})=(x_0,x_{-1},\dots,x_{-k})) \\ =\lim_{\varepsilon\to 0}\lim_{T\to\infty}\frac{\sum_{\tau\in\mathbf{T}_\varepsilon,\tau>-T}\theta_{\varepsilon,U}(\tau)}{|\mathbf{T}_\varepsilon\cap[0,-T]|}. \tag{1.11}$$

The order of limits $\varepsilon\to 0$ and $T\to\infty$ cannot be changed. Practically, this means that for fixed (sufficiently small) ε we have to analyze the behaviour of the fraction $\sum_{\tau\in\mathbf{T}_\varepsilon,\tau>-T}\theta_{\varepsilon,U}(\tau)/|\mathbf{T}_\varepsilon\cap[0,-T]|$ when T grows.

Remark 1.13. We do not assume any Markov condition or hypothesis that the probability of the future is completely defined by a fragment of the past history of a given length. We consider the prediction problem: *we know a fragment of history with some accuracy; evaluate the conditional probability of the future development.*

1.2.3 The kNN sampling for the probability distribution estimation

One of important properties of kNN is its convergence property. Let $x_1, x_2, \dots, x_p$ be observations of dimension p. Consider a point z of dimension p with probability density function f at this point to be $f(x_1, x_2, \dots, x_p)$. Let $k(n)$ be a non–decreasing sequence of positive integers such that:

$$\lim_{n\to\infty} k(n) = \infty, \tag{1.12}$$

and

$$\lim_{n\to\infty} k(n)/n = 0. \tag{1.13}$$

Let the volume of the hypersphere of dimension p about point z with radius r be $A_{r,z}$. The measure of the hypersphere $\hat{f}_n(z)$ is defined as:

$$\hat{f}_n(z) = (k(n)-1)/nA_{r,z}, \tag{1.14}$$

and $\hat{f}_n(z)$ is proved to be consistent (i.e. $\hat{f}_n(z)\to f(z_1,\dots,z_p)$) [55]. This density estimator is proved to be unbiased under assumptions when the sample size is finite [56]. Let L_n be the error of probability that a kNN classifier on random training set and $\hat{L}_n$ be its deleted estimate. For unconditional error of probability R_n, if the feature vector X has a density in R^p and the class probabilities are

continous, it is shown that:

$$n^{\frac{1}{2}}(\hat{L}_n - L_n) \xrightarrow{w} \mathcal{N}(0, \sigma^2), \tag{1.15}$$

and

$$n^{\frac{1}{2}}(\hat{L}_n - R_n) \xrightarrow{w} \mathcal{N}(0, \sigma^2), \tag{1.16}$$

where $\xrightarrow{w}$ represents converge weakly, σ^2 depends on the joint distribution of feature vector and the true class [57].

1.2.4 The kNN sampling for the time series prediction problem and historical Monte–Carlo

As we can see, selection of k nearest neighbors is equivalent to selection of ε–vicinity in the following sense. Consider a probability distribution in R^n and a condition $f(x) = a$, where f is a continuous map from R^n to R^q. Assume that we have to find the conditional distribution on a preimage a for variable a. This means that for every open set $U \subset R^n$ we have to find the probability $P(X \in U | f(X) = a)$. For every $\varepsilon > 0$ we can evaluate the probabilities $P(f(X) \in B_\varepsilon^q + a)$ and $P(X \in U \& f(x) \in B_\varepsilon^q + a)$. For this purpose, let use find N values of X in independent trials. Let $\#(f(X) \in B_\varepsilon^q + a)$ be the number of those values of X for which $f(X) \in B_\varepsilon^q + a$ and let $\#(X \in U \& f(x) \in B_\varepsilon^q + a)$ be the number of such values of X that $f(X) \in B_\varepsilon^q + a$ and $X \in U$. Assume that $\#(f(X) \in B_\varepsilon^q + a) > 0$. Then the relative frequency gives the frequentist estimate of conditional probability:

Proposition 1.14. *With probability one*

$$P(X \in U | f(x) = a) = \lim_{\varepsilon \to 0} \lim_{N \to \infty} \frac{\#(X \in U \& f(x) \in B_\varepsilon^q + a)}{\#(f(X) \in B_\varepsilon^q + a)}.$$

This is just the law of large numbers combined with the definition of conditional probability. The kNN sampling (instead of ε vicinity) also gives the estimate of conditional probability. Take a positive integer k. Let us find N values of X in independent trials, $N > k$. Select from this N values k values, for which $f(X)$ are the closest to a. (Order empirical values of X in the order of the distance $\rho(f(X), a)$, from smallest to largest; if there are several samples with the same value of distance then order then according to the number of the trial). Select from this k nearest neighbor samples those, which belong to U. Let their number be $\#(\text{kNN's} \in U)$.

Proposition 1.15. *With probability one*

$$P(X \in U | f(x) = a) = \lim_{k\to\infty} \lim_{N\to\infty} \frac{\#(kNN's \in U)}{k}.$$

This is the version of the law of large numbers with kNN sampling.

Using the kNN version of the law of large numbers we can formulate the kNN sampling rule for prediction of the future for stationary ergodic processes. Let X_t $(t \in Z)$ be a stationary ergodic random process, x_t $(t \in Z)$ be a trajectory of this process. Select a piece of history of length $q+1$, $X_0 = x_0, X_{-1} = x_{-1}, \ldots, X_{-q} = x_{-q}$, and the depth of prediction m.Let us use for prediction and selection of nearest neighbors the fragment of the past trajectory x_τ $(-N-q-m \leq \tau < -m)$. For each τ 'the future of length m' is defined as $(x_{\tau+1}, x_{\tau+2}, \ldots, x_{\tau+m})$ and 'the past of the length $q+1$' is defined as $\pi_\tau = (x_\tau, x_{\tau-1}, \ldots, x_{\tau-q})$. Select from the set $\{\pi_\tau \mid -N-q-m \leq \tau < -m\}$ k nearest neighbors to the known piece of history $(x_0, x_{-1}, \ldots, x_{-q})$. The ensemble of the futures of length m corresponding to these kNNs gives the sample that predicts the future distribution. This ensemble is the set of k sequences of length m. Let us denote him $M_{k,N}$.

Proposition 1.16. *For every open subset $U \subset R^m$ with probability one:*

$$P((X_1, X_2, \ldots, X_m) \in U | X_0 = x_0, X_{-1} = x_{-1}, \ldots, X_{-q} = x_{-q} = \lim_{k\to\infty} \lim_{N\to\infty} \frac{M_{k,N}}{k}.$$

The order of limits $k \to \infty$ and $N \to \infty$ cannot be changed. Practically, it means that for fixed k we have to analyze behavior of the fraction $\frac{M_{k,N}}{k}$ when N grows and then increase k keeping the condition $k \ll N$.

1.2.5 The efficient market hypothesis

There are many researchers believe that the historical data contains some information about future while some others believe that the stock prices react as a random walk. The Efficient Market Hypothesis (EMH) is introduced in 1969, evidence shows that the new information of the split would effect on the share prices at the end of a split month or maybe instantaneously after the announcement date [58]. In general, three major forms of this hypothesis are discussed over decades, the weak–form, semi–strong form and strong–form efficiencies. The weak–form suggests that the future prices could not be predicted by analysis of

historical prices in any form while for the strong–form, the share prices contain all information that it is impossible to have an opportunity to earn excess returns. Many critics of this hypothesis are discussed in 1970s–1980s to discuss the 'efficiency' of the market such that if a market is efficient with respect to some information, then the prices fully reflected this information [59] [60]. Studies on testing efficient market hypothesis of European stock and share markets for selected time period shows that for 6 selected European countries, Germany market has the highest efficiency [61].

1.2.6 The technical analysis

Technical analysis is applied by many traders and investors to discover trends and patterns for historical time series (i.e. stock and share daily closing prices) to forecast the future direction of time series. By analyzing the charts and developing indicators, technical analysis can assist traders in building an efficient trading strategy. The assumption of technical analysis contradicts the Efficient Market Hypothesis (EMH). The assumption of research for technical analysis believes that strategy could be built to gain consistent profit [62]. The application results of technical analysis vary according to types of financial time series and different regions of markets. Survey of technical analysis on foreign exchange market shows that there are significant amounts of dealers considered technical analysis as complementary analysis [63]. The study on the results of simulating four technical trading rules over the 1960 to 1983 period indicates that it is too difficult for technical analysis to forecast subsequent prices [64]. Filter rules constructed from the technical analysis are used on intra–daily foreign exchange market. These rules can generate some profits but not for the general case [65]. Many surveys study about profitability, theorems and empirical work regarding technical trading strategies are reviewed but they cannot agree on a consistent profitability of technical analysis [66]. The positive profitability of technical analysis could imply that historical prices contain information about future prices. Another interesting analysis of 20 new equity markets in emerging economies shows that the correlation is lower for the developed country returns [67] in other words, for markets in developed countries, it is more difficult to gain profit by using technical analysis. Studies of using decomposition–based vector autoregressive model had shown that the significant information that technical analysis provides for FTSE100 index between 1984–2002 [68]. Another study of technical analysis has partially prediction

power to US equity index returns [69]. Technical analysis and nearest neighbor technique are combined to construct an automatic trading rule and it makes higher profit than general buy–and–sell strategy [70].

The Efficient Market Hypothesis and profitability of technical analysis are leading to an interesting question: does historical price contains information about the future price? To answer this question, we developed 2 different experiments using idea of kNN. In the first experiments, a label system is developed for components of selected markets to label components with 'winner' or 'loser'. Then apply Leave–One–Out Cross–Validation of kNN algorithm to label the components with either 'winner' or 'loser'. From the results of error analysis and visualization of data using ViDaExpert, it shows that for HANGSENG index, there is a positive predictability while for DAX index, FTSE index and NASDAQ index, there is only limit predictability. Hence, technical analysis could be applied to the market with positive predictability. In the second experiment, we propose a universal approach based on developing experiments using kNN to test the predictability on both ergodic dynamical systems (EMH assumption) and stationary ergodic random process (technical analysis assumptions) for selected stock and share markets over a time period.

1.3 Outline of thesis

The main topic of my Ph.D research is to develop models and experiments using kNN to analyze the predictability for stock and share time series. In Chapter 1, generalizations, modifications and applications of NN and kNN algorithms are reviewed. The NN rules are split into 2 different types: structure–less and structure–based methods. In this thesis, the basic structure–less kNN algorithm is mainly applied. The motivation of this thesis is also discussed with reviews on technical analysis, the predictability of time series and efficient market hypothesis. Some background knowledge for time series, random process, ergodicity and stationarity are reviewed as a support for the motivation.

In Chapter 2, a case study is performed with four stock and shares indices within a 3–year time period between 02–07–2009 and 02–07–2012. After labeling the component with 'winner' or 'loser' by selecting first 1/3 percent and last 1/3 percent of sorted average price ratios. Components labeled with 'winner' are those who has 1/3 smallest average prices ratios. This means that the mean price of the

final time period is relatively higher than the beginning time period. Components labeled with 'loser' are those who have the 1/3 largest average prices ratios. It means they have higher mean prices of beginning time period. The information about 'winner'–'loser' separation in the initial fragments of the daily log–returns time series can be shown in error analysis of Leave–One–Out Cross–Validation. Leave–One–Out Cross–Validation with kNN algorithm are applied on the daily log–return of components using a distance and proximity in the experiment. By looking at the error analysis, we see that for the HANGSENG and DAX indices, there are clear signs that one can evaluate the probability of long–term success. The correlation distance matrix histograms and 2–D and 3–D elastic maps generated from ViDaExpert show that the 'winner' components are closer to each other and 'winner' \'loser' components are separable on elastic maps for HANGSENG and DAX index while for the negative possibility indices, there is no sign of separation.

In Chapter 3, we develop a kNN based universal technique or 'historical Monte Carlo' method which allows one to predict and model stationary ergodic time series. For selected stock and share closing prices, the daily log–returns are computed for the time series. The log–returns are split into 3 parts, the "history" part, the "present" part and the "future" part. The idea of this approach is to apply kNN rule for "present" part log–return time frame, search for k nearest neighbor time frame from the training set. These k nearest neighbors are then used as log–return factors for predict "future" part. Then the log–return factors are transformed into closing prices using the concept of dynamical systems. There are four measurements of distances applied in the experiment, Euclidean distance, City Block distance, correlation distance and cosine similarity. A different training set of "history" part could change the prediction results. By analyzing 3 extreme cases on different data, the predicted results are close to the real data. The large training set would lead to a better result however this would increase the storage of distance matrix. A comparison between the ARMA(1,1) and the kNN experiment is made and it has been shown that the prediction results from the kNN experiment and the ARMA(1,1) are similar. Hence, we conclude that this approach works well for both ergodic dynamic systems and for various types of random process. A Lorenz system is a deterministic and chaotic system of differential equations. Data generated from the Lorenz attractor with a random noise is used to test the performance of the kNN experiment. From the analysis of average RMSE and average standard deviations, it shows that average RMSE converges as $k \to \infty$.

When $k = 1$, average RMSE can be very small. This means that the best k is acheived when $k = 1$. Chapter 4 includes the conclusion and future research directions of our research program.

Chapter 2

Is it possible to predict long–term success with kNN? Case Study of four market indices (FTSE100, DAX, HANGSENG, NASDAQ)

2.1 Introduction

Prediction of time series is an essential and difficult task in the real world. Many methods have been studied these days such as constructing complex models for simulating the prices, different types of regression models and so on. It is important to study the predictability of the time series separately from constructing the models because in the creation of each model we assume some additional hypotheses about the model structure. It is also interesting to study this problem as when there is a positive predictability or sign of changes in the time series, traders may use this as a sign to discover crisis to prepare a response for some critical situations. For example, during crisis, the correlation and variance is higher [71][72].

In this work, we test the possibility of prediction of long–term success on the financial market. The time interval is three years. We evaluate the success of the companies during these three years. The main question is: was there a similarity between the most successful companies and dissimilarity between them and the least successful ones at the beginning of this time period? In other words, is there anything in the previous history that may give us information about the success

in the following three years? We use movement of prices only and our study should demonstrate the possibility of using historical data for long–term success prediction.

Our case study is aimed to find the likelihood of the prediction of "success" within three years' time interval from 02–JUL–2009 to 29–JUN–2012 for four selected stock and share market indices. We compare their performance in the two time frames: initial three–month frame at the beginning (02/07/2009–30/09/2009) and the final three– month frame (02/04/2012–29/06/2012). The idea of the main experiment is based on the backward analysis. Backward analysis can be defined as an analysis to determine properties of the inputs of a program from properties or contexts of outputs. This case study is aimed to construct experiments on the data to test if it is possible to predict long–term success. The possibility of long–term "success" of the selected indices is tested from the results of the experiment in the three years' time interval. For each stock market index, the closing prices of all components are collected from Yahoo! Finance website. After data pre–processing step, the remaining components are labeled with "winner" companies or "loser" companies (or simply just "winner", "loser") by using the "1/3 average price" approach. For this approach, we compute the average price ratio which is defined as the mean price of the end period divided by the mean price of the beginning period. The companies are then sorted by the descending order of this computed ratio. We label the first 33.3% of companies as "winner" and label the last 33.3% of the companies as "loser". Then the log-return prices are computed on this data. The kNN algorithm with Leave-One-Out Cross-Validation with two distance measurements is used as the indicator to test the possibility of prediction.

We use Leave–One–Out Cross–Validation for kNN classifier to test the possibility of prediction of "success" components for each market index. The data is collected from data and cleaned. Then we did an experiment of Cross–Validation for kNN using two different forms of distance measurements. Then we analyze the total error and separate error and use two methods to visualize our result. We show that there is a possibility of predictions for long–term success for HANGSENG and DAX indices in Section 2.3. We summarize our result and conclude in Section 2.4.

2.2 Methods and backgrounds

2.2.1 Date pre–processing

The closing price is the final price at which a security (in this case study, the stock exchange) is traded on a given trading day. It represents the value of this security on a trading day until it changes again on the next trading day. The raw data used for the experiments of this paper are the closing prices of all components for each index for different time frames (3 months, 6 months, 9 months, 12 months, 15 months and 18 months). These closing prices are collected from Yahoo! Finance. The first cleaning step is to compare the dates of each company with the index's date. The prices on the dates that are not in the trading date of the index are deleted. The second step is dealing with the missing values. Companies which have more than 20% of missing values are deleted from the list of companies for further experiment. For the rest of companies with missing values, the missing values are filled with the attribute mean (mean of the closing price on a specific date). The closing prices are sorted from the oldest to most recent and they are saved in a matrix with each column represents each company and each row represents each date.

2.2.2 The Log–return

The kNN algorithm is applied using the next day's daily log–return of closing prices. There are two main reasons for using returns. First, for average investors, the return of an asset is a complete and scale–free summary of the investment opportunity. Second, return series are easier to handle than prices series as they have more attractive statistical properties [73]. Let P_t be the closing price of each company at time t. The log–return at time t is defined as:

$$r_t = \ln \frac{P_t}{P_{t-1}} \tag{2.1}$$

Hence, if consider the log–returns as a matrix defined as the form of closing prices, the matrix of log returns will have one less row than the matrix of closing prices. The advantages of log returns over the simple net returns are obvious. First, the log return is the sum of continuously compounded one-period returns involved. Second, statistical properties of log returns are more tractable.

2.2.3 The Pearson's product–moment correlation coefficient

Two types of distance measurements are used in kNN algorithm in this case study. They are distinguished as different functions of correlation coefficients. The correlation coefficient measures the dependence of two random variables. The Pearson's Correlation or Pearson's product–moment coefficient is defined as the covariance of two random variables divided by the product of the individual standard deviations. For a series of n measurements of X and Y written as x_i and y_i , where $i = 1, 2, \ldots, n$. The correlation coefficient is defined as:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{2.2}$$

The coefficient is bounded between $+1$ and -1. When coefficient equals to 0, it means there is no linear relationship between X and Y. When the coefficient equals $+1$, it means there is a positive linear relationship between X and Y. When it equals -1, it means there is a negative linear relationship between X and Y.

2.2.4 The Correlation distance

In cluster analysis, the correlation distance is used in a specific metric. The correlation distance on returns of time series $(\boldsymbol{X}, \boldsymbol{Y})$ is defined as:

$$d(\boldsymbol{X}, \boldsymbol{Y}) = \sqrt{2(1 - c_{\boldsymbol{XY}})} \tag{2.3}$$

where $\boldsymbol{X} = (x_1, x_2, \ldots, x_\tau)$, $\boldsymbol{Y} = (y_1, y_2, \ldots, y_\tau)$ and $c_{\boldsymbol{XY}}$ is the correlation coefficient between $\boldsymbol{X}$ and $\boldsymbol{Y}$ [74]. It is used as in the analysis of a case study for the Italian hospitality sector [75]. Proximity of this distance measurement is computed as:

$$d(\boldsymbol{X}, \boldsymbol{Y}) = \sqrt{1 - c^2_{\boldsymbol{XY}}} \tag{2.4}$$

where $\boldsymbol{X} = (x_1, x_2, \ldots, x_\tau)$, $\boldsymbol{Y} = (y_1, y_2, \ldots, y_\tau)$ and $c_{\boldsymbol{XY}}$ is the correlation coefficient between $\boldsymbol{X}$ and $\boldsymbol{Y}$. Consider the geometric interpretation of the correlation coefficient, it can be thought as the cosine of the angel between $x_i - \bar{x}$ and $y_i - \bar{y}$ where $x_i \in \boldsymbol{X}$ and $\bar{x}$ is the mean of all x_i, $y_i \in \boldsymbol{Y}$ and $\bar{y}$ is the mean of all y_i. Using double formulas of cosine, the distance function can take another expression $2\sin\alpha$ where $\alpha \in [0, \frac{\pi}{2}]$.

$$2\sin\alpha = \sqrt{2(1 - 2\cos\alpha)} \tag{2.5}$$

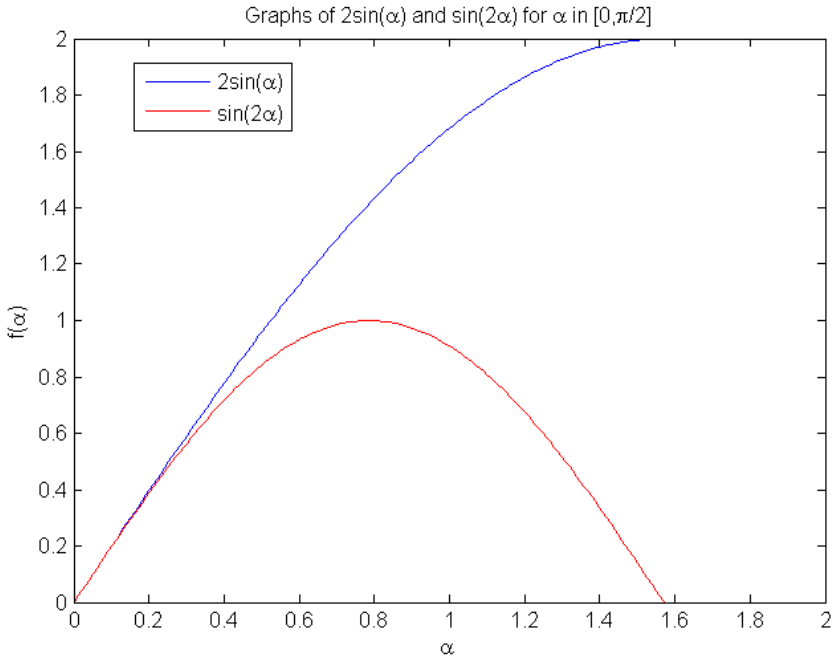


FIGURE 2.1: Graph of $2\sin(\alpha)$ (Distance) and $\sin(2\alpha)$ (Proximity of Distance) for $\alpha \in [0, \frac{\pi}{2}]$.

The distance $d(\boldsymbol{X}, \boldsymbol{Y})$ can have another expression of $2\sin\alpha$. The proximity is defined in the similarly way, with an expression of $\sin(2\alpha)$:

$$\sin(2\alpha) = \sqrt{1 - \cos^2(2\alpha)} \tag{2.6}$$

This expression is 0 when $\alpha = 0$ and $\alpha = \frac{\pi}{2}$.

The comparison of these two expressions is shown in Figure 2.1. These two expressions are the same when α is small and grows linearly as α increases. But the plot of proximity starts to reach its maximum at 1 when α is approaching to 0.8 approximately and then it starts to decreases and reaches zero again at around $\alpha = 1.6$. This small difference has almost no effect on the later experiments.

2.2.5 The average price and the long term success

Before applying the kNN algorithm, the training data (data with labels) are required. The "1/3 average method" is used to label the components (i.e. label them with either "winner" or "loser"). The "winner" companies are the companies that have a relatively higher average price in the last time frame while the "loser" companies are the ones have a relatively higher average price in the beginning time

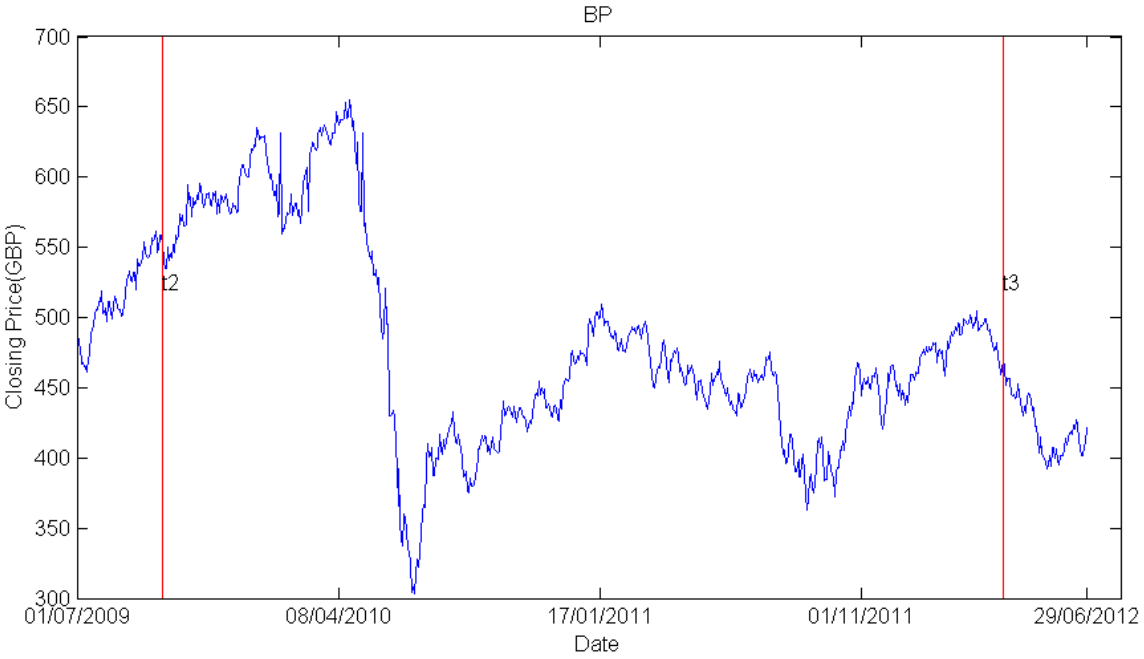


FIGURE 2.2: Graph of company BP, component of FTSE100 index with date against its closing price in GBP between 01/07/2009 and 29/06/2012.

frame. The average price of the time series in the specific time period is defined as the sum of the total prices in this period divided by the number of prices. (i.e. if in a time period there are n prices, each specific price in this period is denoted as $P_1, P_2, \ldots, P_n$, the average price $Avp = \frac{1}{n}\sum_{i=1}^{n} P_i$).

For example, Figure 2.2 is a time series of a component of FTSE100 index. The red lines labeled with 't2' and 't3' represent the two border lines of the beginning and last 3 month time period. The average prices of the first and last time period are calculated using the definition introduced earlier and let Avp_1 be the average price for the first three months, and let Avp_2 be the average price for the last three months. The ratio of the average price of this company is then defined as the ratio of $Avp1$ and Avp_2. (i.e. $ratio = \frac{Avp1}{Avp2}$) The next step of "1/3 average price" method is to sort this ratio in descending order. The first $\frac{1}{3}$ companies are labeled as "winner" while the last $\frac{1}{3}$ companies are labeled as "loser".

The Nearest–Neighbors (NN) predictors were studied in several papers. The NN predictor is applied for the analysis of forecasting daily exchange data in foreign exchange markets [76]. This paper gives an interesting application of NN in financial time series. A similar study, using Simultaneous Nearest–Neighbor (SNN) predictors is, applied to nine EMS currencies using daily data [77]. Hence, it is interesting to use k-Nearest Neighbor as an indicator to test predictability.

2.2.6 The kNN algorithm and the LOOCV

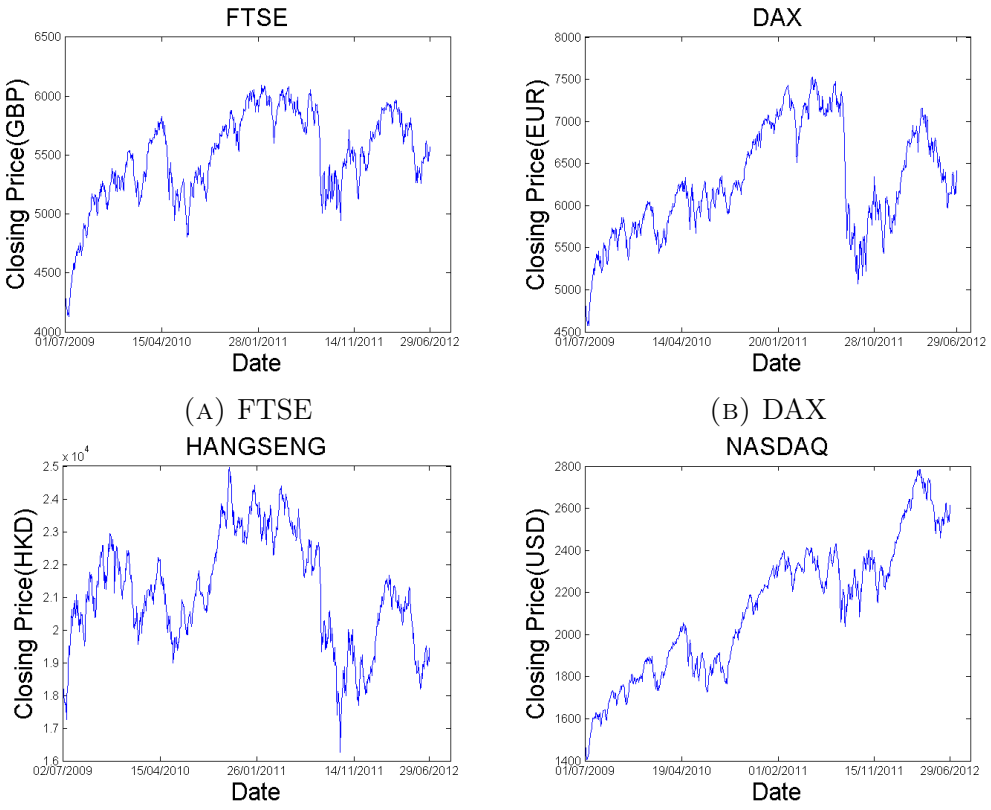
The k–Nearest–Neighbor Classifier is one of the famous and simplest classification algorithms. It requires no models to fit [67]. The classification rule is for a specific test point with no label, it can be classified using majority vote among the k–Nearest–Neighbors. The nearest neighbors of a test point are found by looking for the k smallest distances between the test point and the training points. There are many potential distance functions. One of them is the Euclidean distance in the feature space:

$$d_{(i)} = ||x_{(i)} - x_0|| \tag{2.7}$$

where x_0 is a test point and $x_{(i)}$ are points in the training set. In this Chapter, the distances used will be the two distances introduced earlier. To estimate the prediction error, cross–validation is used. This is probably the simplest and most widely used method when we do not have enough amount of data [78]. This predictor is also The K–fold cross–validation uses part of the data to fit the model, in my experiment, the kNN classifier, and a different part to test it. In K–fold cross–validation, the data is split into K equal–sized parts. For the Kth part, when using a model to predict the Kth part of data, the $K-1$ parts of data are used to fit the model. The leave–one–out cross–validation is a special case when K is the total number of data points. When applying this method to kNN, it extracts one point from the original data and this point is considered as a test point. Then the rest of them are used as the training points.

The time period of the experiments is chosen between 01/07/2009 and 29/06/2012 (For some stock markets, it may vary as it might not have prices at 01/07/2009 or 29/06/2012, and the next trading date will be counted as the boundary). Because of the financial crisis occurred in 2007/2008, the stock and exchange indices time series were affected.

Figure 2.3 shows the movement of the closing prices of four market indices between 01/07/2009 and 29/06/2012.The closing prices are relatively low in the beginning for all markets. Then they begin to increase and except for HANGSENG index, they have a generally up trend. This uptrend stands for the recovery of the market from the crisis. For HANGSENG index, the closing price was increasing then at some date, it starts the downward trend and it was likely to be hit by the second wave of crisis.



(A) FTSE (B) DAX

(C) HANGSENG (D) NASDAQ

FIGURE 2.3: Figures of market index in the three years period between 01/07/2009 and 29/06/2012 (For HANGSENG index, it does not have price on 01/07/2009 hence it starts from the next trading date) (a) Closing price of FTSE (b) Closing price of DAX (c) Closing price of HANGSENG (d) Closing price of NASDAQ

2.2.7 Estimation of proportion

In statistics, sampling theory studies the relationship between the population and sample from this population. Estimate of proportion is considered as an interval estimate of the population proportion [79]. Suppose that a population is infinite and the probability of occurrence of an event is p , and consider all possible samples of size N drawn from this population, a sampling distribution of proportions with mean μ and standard deviation σ is given by:

$$\mu = p \text{ and } \sigma = \sqrt{\frac{p(1-p)}{N}} \tag{2.8}$$

In Section 2.3, the sampling distributions of "winner" and "loser" companies are computed. It gives more information in the case if the testing number of companies is small.

Index Name	Total	Used	Deleted	Winner	Loser
FTSE100	101	98	3	32	32
DAX	30	30	0	10	10
HANGSENG	50	49	1	16	16
NASDAQ	100	100	0	33	33

TABLE 2.1: Table of Number of Companies contained in the experiment for specific market index. "Total" means the total number of components (companies) in the index. "Used" means the number of companies left after the data pre–processing step. "Deleted" means the number of companies deleted in the data pre-processing step. "Winner"/"Loser" means the number of companies are labelled with "Winner"/"Loser".

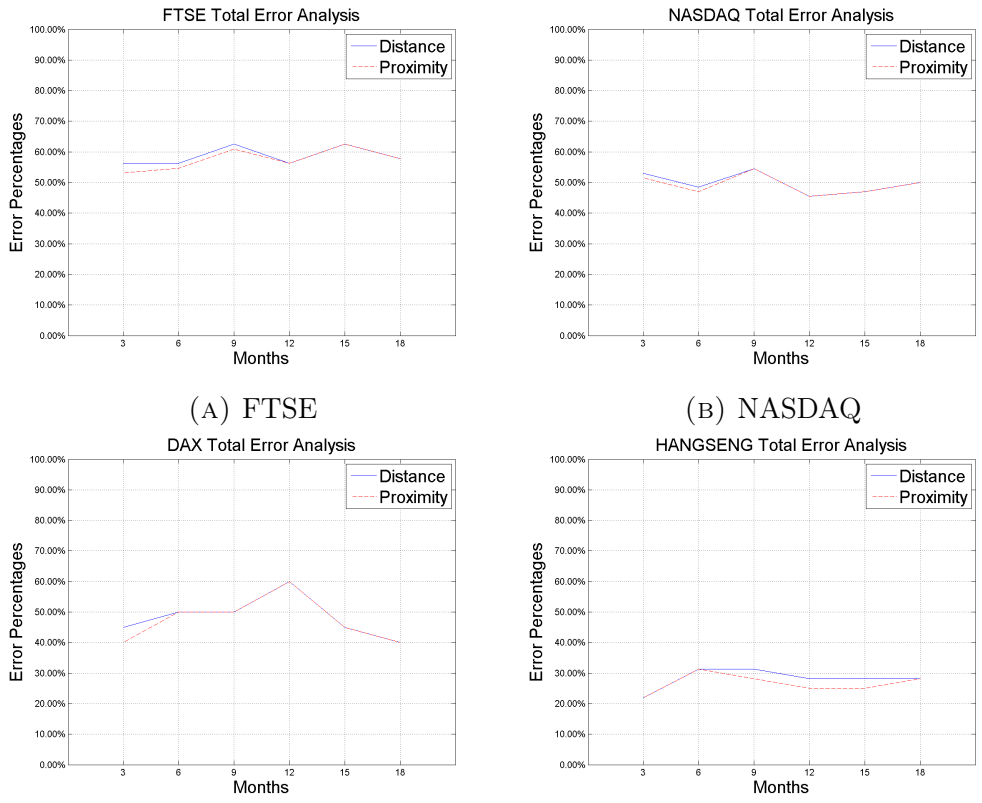
2.3 Results and analysis

The experiment is computed using MATLAB. It begins with the data pre–processing step. The companies that do not have enough amounts of closing prices are deleted from the company list.

From Table 2.1, the number of deleted companies is relatively small compared to the total number of components. The next step is to apply the "1/3 Average Price" method for the remaining companies, labels them with "winner" if the corresponding average prices are the largest 1/3 of the sequence of average prices, and "loser" if the average prices are the last $\frac{1}{3}$ of the sorted sequence. The resulting data is generated by joining the "winner" and "loser" companies in matrix form vertically. In this matrix, each column represents each date and each row represents each company. The daily log-return matrix is computed from the joint matrix. This log-return matrix is then used for leave–one–out cross–validation of 1NN algorithm.

2.3.1 The analysis of total error and separate error

The error of leave–one–out cross–validation for 1NN is performed for different time periods. (i.e. from 3 months to 18 months) The total error is referred to the number of misclassified points for both "winner" and "loser" companies. Figure 2.4 shows the results of total error analysis for different indices. The errors generated using different functions of measurement are almost identical. The proximity can generate a bit smaller error than the distance function for several months. The total error for the time period of 3 months is the minimum for three indices. For



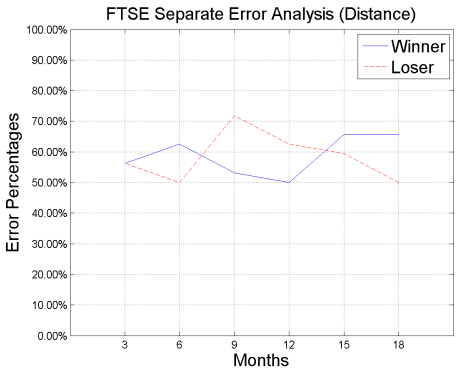
(A) FTSE (B) NASDAQ

(C) DAX (D) HANGSENG

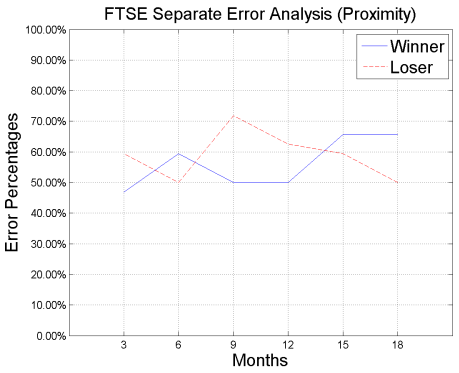
FIGURE 2.4: Figures of total error analysis of leave–one–out cross–validation for 1–NN results from 3 months initial time period to 18 months initial time period for different markets (a) FTSE index (b)DAX index (c) HANGSENG index (d) NASDAQ index

indices FTSE, DAX and NASDAQ, the errors are mostly around 50%. This means the prices are random and there is no sign of the possibility of prediction. For HANGSENG index, the error percentage is less than 50%. This shows that this market is not completely random. To analyze the characteristics for "winner" and "loser" companies, a separate error analysis is applied. The sum of "winner" error number and "loser" error number should be the same as the total error number.

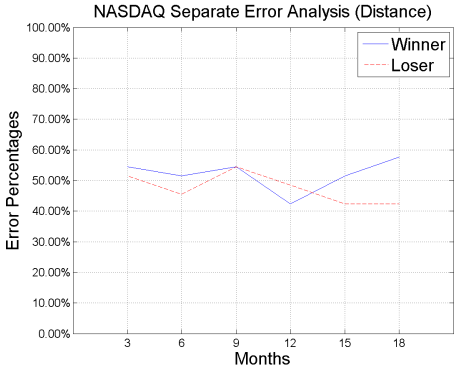
The result of this analysis is shown in Figure 2.5. For indices FTSE and NASDAQ, the error percentages are around 50% for both "winner" and "loser" companies. Hence, the separate error analysis shows that there is no sign of the possibility of prediction. For the DAX index, the errors range from 30% to 60%, it can be seen that a border case and the "loser" companies, in general, have slightly lower errors than "winner" companies. For the HANGSENG index, both errors are below 50% and the errors of "winner" companies are much smaller than the errors of "loser" companies. Hence, this means for HANGSENG index, there are some conclusions about predictability for the "winner" companies.



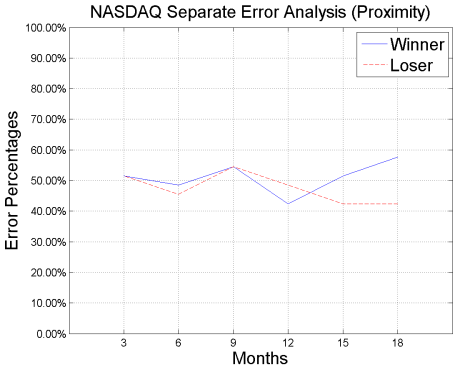
(A) FTSE (Distance)



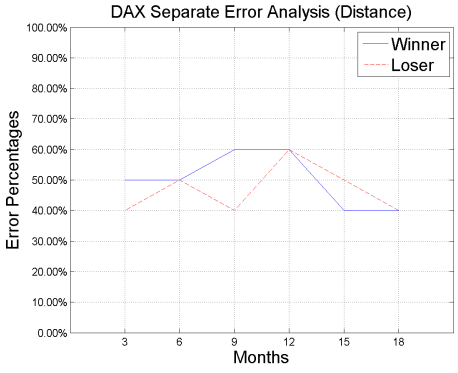
(B) FTSE (Proximity)



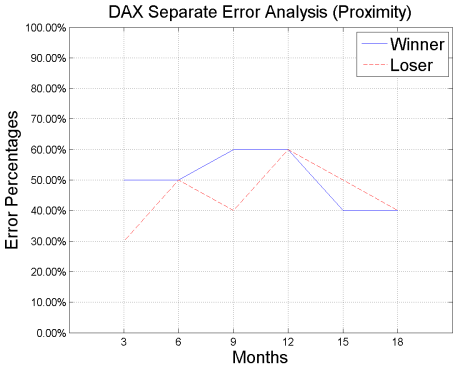
(C) NASDAQ (Distance)



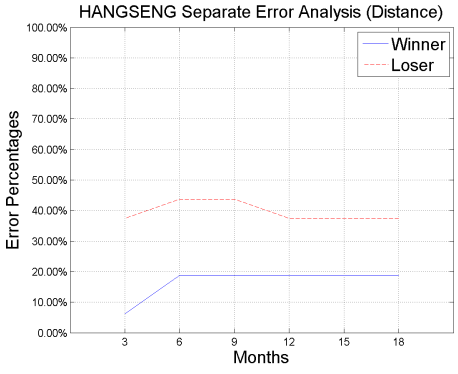
(D) NASDAQ (Proximity)



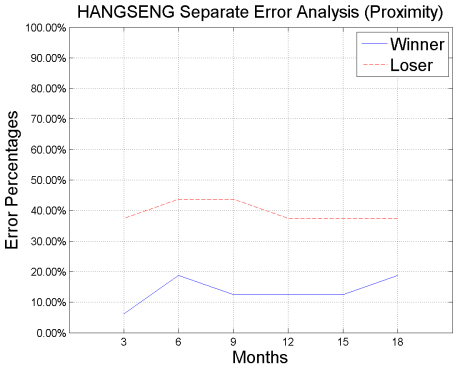
(E) DAX (Distance)



(F) DAX (Proximity)

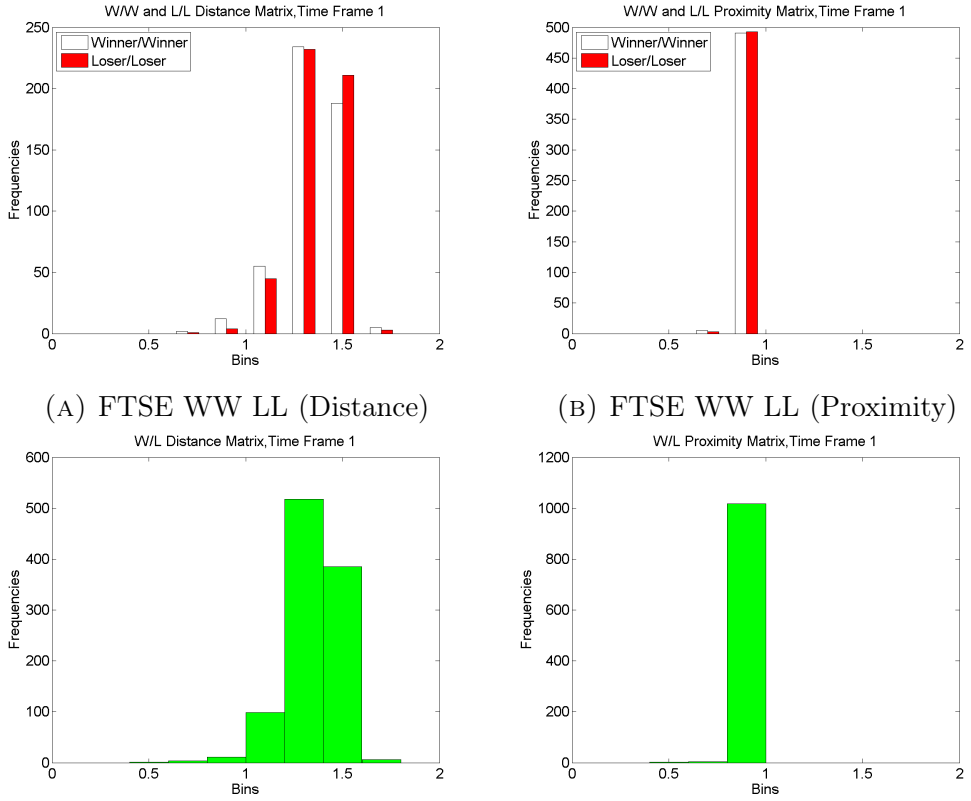


(G) HANGSENG (Distance)



(H) HANGSENG (Proximity)

FIGURE 2.5: Figures of separate error analysis of leave–one–out cross–validation for 1–NN results from 3 months initial time period to 18 months initial time period for different markets.



(A) FTSE WW LL (Distance)

(B) FTSE WW LL (Proximity)

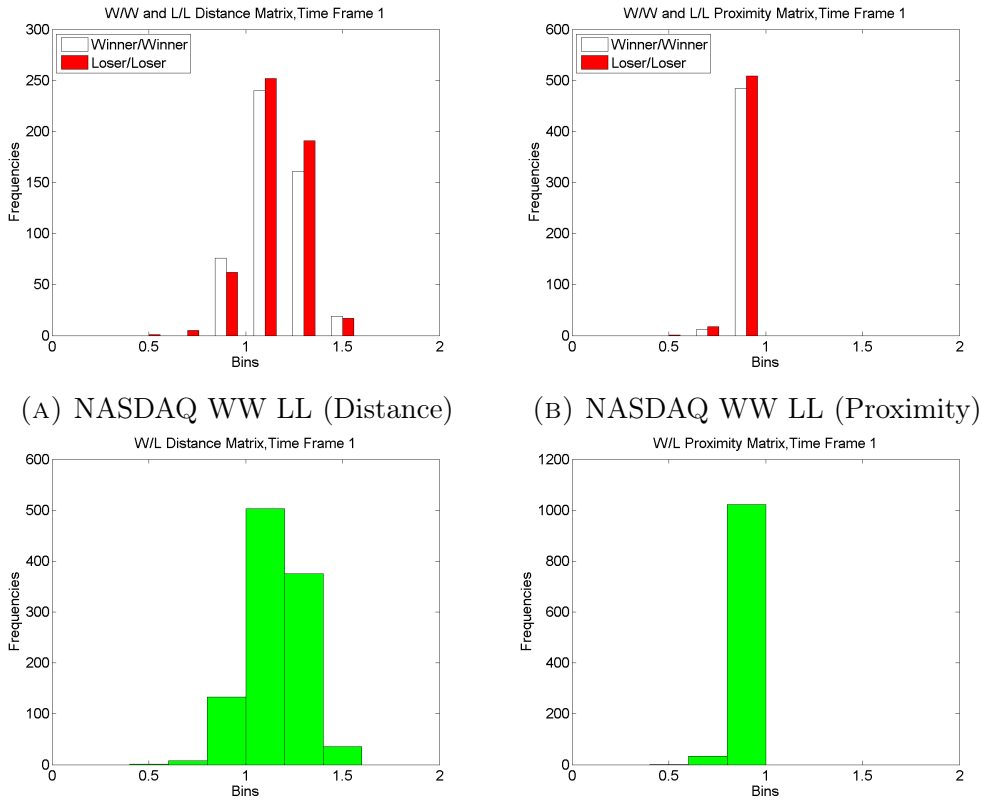
(C) FTSE WL (Distance)

(D) FTSE WL (Proximity)

FIGURE 2.6: Histograms of different expressions of correlation distances for FTSE index when using first 3 months closing prices for experiment. (a) Using Distance between winner/winner and loser/loser companies (b)Using Proximity between winner/winner and loser/loser companies (c) Using Distance between winner and loser companies (d) Using Proximity between winner and loser companies

2.3.2 Visualization using the histograms of the correlation distance matrix

The results from Figure 2.4 and Figure 2.5 shows the minimum error rate occurs when the time interval is within 3 months for FTSE, DAX, and HANGSENG index. The sign of the possibility of prediction in this time period is most dominant among all. The histograms of correlation distances can be used to study the distribution of correlation distance. For each index, four histogram plots are generated. They represent the distributions of in–class (winner/winner or loser/loser) correlation distances and cross–class (winner/loser) correlation distances. In Figure 2.6, Figure 2.7, we have histograms for FTSE and NASDAQ indices. They have no signs of possibility of prediction.



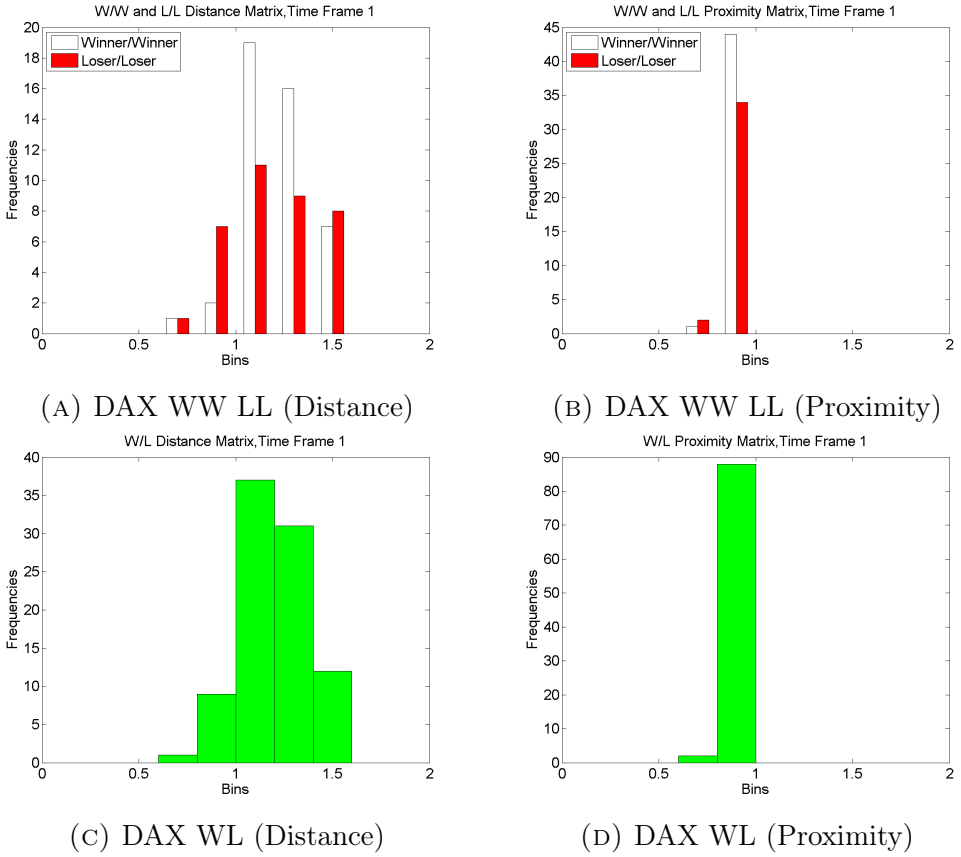
(A) NASDAQ WW LL (Distance) (B) NASDAQ WW LL (Proximity)

(C) NASDAQ WL (Distance) (D) NASDAQ WL (Proximity)

FIGURE 2.7: Histograms of different expressions of correlation distances for NASDAQ index when using first 3 months closing prices for experiment. (a) Using Distance between winner/winner and loser/loser companies (b)Using Proximity between winner/winner and loser/loser companies (c) Using Distance between winner and loser companies (d) Using Proximity between winner and loser companies

Figure 2.5 shows the result of LOOCV error analysis for "winner" and "loser" companies separately. For the figures on the same row, these represent the result of the experiment using companies from the same market but different expressions of distance (use Distance or Proximity). (a)FTSE index use Distance expression (b)FTSE index with Proximity expression (c)DAX index use Distance expression (d)DAX index with Proximity expression (e)HANGSENG index use Distance expression (f)HANGSENG index with Proximity expression (g)NASDAQ index use Distance expression (h)NASDAQ index with Proximity expression.

The in–class histograms show that the distribution of distances between "winner" themselves and "loser" themselves are almost identical. The distributions of in–class distances and cross–class distances are quite similar as well. This makes "winner" and "loser" companies not easily separated. Since kNN is an algorithm which is sensitive to the data structure. If the points are mixed together, this



(A) DAX WW LL (Distance)

(B) DAX WW LL (Proximity)

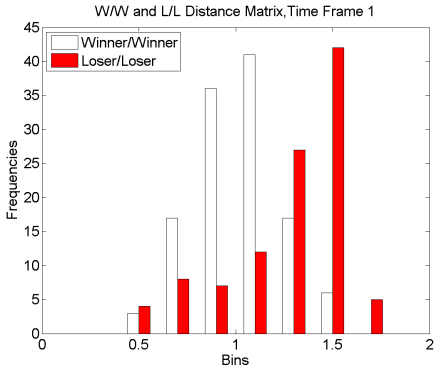
(C) DAX WL (Distance)

(D) DAX WL (Proximity)

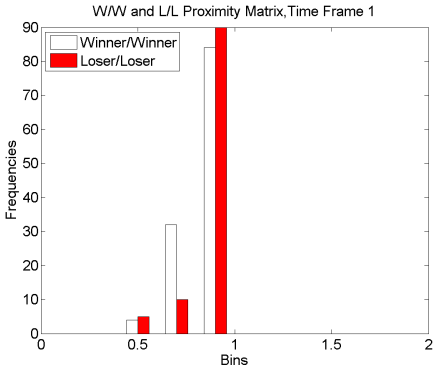
FIGURE 2.8: Histograms of different expressions of correlation distances for DAX index when using first 3 months closing prices for experiment. (a) Using Distance between winner/winner and loser/loser companies (b)Using Proximity between winner/winner and loser/loser companies (c) Using Distance between winner and loser companies (d) Using Proximity between winner and loser companies

algorithm generates errors. Hence, if "winner" and "loser" companies are mixed together, this market has no sign of the possibility of prediction. Figure 2.8 is the histogram of correlation distances for the DAX index.

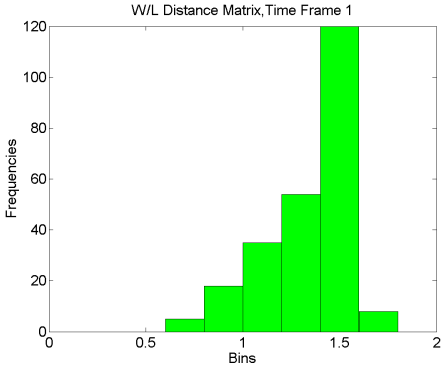
The DAX index has some possibility of prediction but this sign is not very clear. From the in-class distributions, the "winner" and "loser" seems to be separated, but this separation is not clear enough since the distribution peak of winner/winner distances is slightly shifted to the left– hand side of the distribution peak of loser/loser distances. Figure 2.9 shows the distributions of correlation distances for the HANGSENG index. For this index, it has a clear sign of the possibility of prediction. The distribution of winner/winner distances is on the left–hand side of the distribution of loser/loser distances. Hence, the "winner" companies are more compact than "loser" companies. There is a good separation between "winner" and "loser" companies.



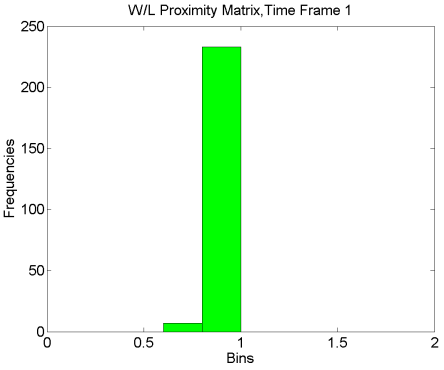
(A) HANGSENG WW LL (Distance)



(B) HANGSENG WW LL (Proximity)



(C) HANGSENG WL (Distance)



(D) HANGSENG WL (Proximity)

FIGURE 2.9: Histograms of different expressions of correlation distances for HANGSENG index when using first 3 months closing prices for experiment. (a) Using Distance between winner/winner and loser/loser companies (b)Using Proximity between winner/winner and loser/loser companies (c) Using Distance between winner and loser companies (d) Using Proximity between winner and loser companies

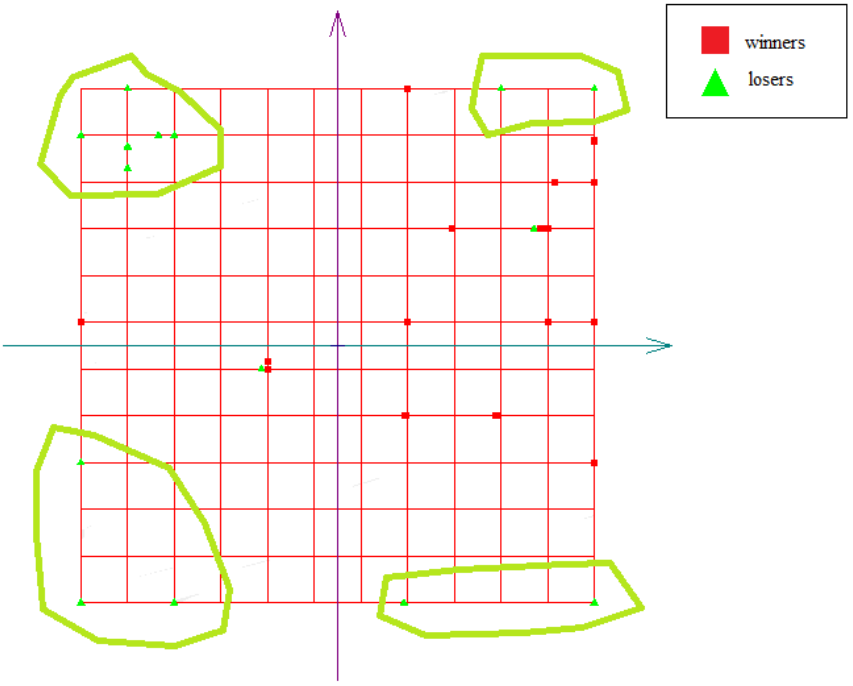
The figures of companies are used to visualize this characteristic. The most difficult part of plotting is the dimension of the dataset. For this data, the dimension is relatively high. (i.e. each date is a dimension) The principal graphs and manifolds can be used to produce the plots with lower dimension. In previous studies, the metaphor of elastic membrane and plate is used to construct one–, two– or three–dimensional principal manifold approximations of various topologies. The mean squared distance approximation error and the elastic energy of the membrane together formed a functional to be optimized [80]. This idea of using elastic graphs is demonstrated on several practical examples: from comparative political science, data analysis in molecular biology and analysis of dynamical systems for biochemical modeling [81]. The software "ViDaExpert" [82] developed by Dr. Andrei Zinovyev uses this idea of elastic energy to compute elastic map and net using the principal manifolds. This software enables users to visualize multidimensional data with the idea of using principal object to reduce the dimension of this data.

In our experiment, we use this software to visualize the log-return prices for all companies within a time period of the initial first 3 months. There are two parameters of this elastic map algorithm to generate the graph, the coefficients of "stretching elasticity" λ and the "bending elasticity" μ . To have the good performance approximation for the principal manifold, I fixed $\lambda \approx 0$ and $\mu \approx 8.1$. For each index, two types of graphs are generated using ViDaExpert. The first one is the "2D–Elastic Map". It displays the estimation of 2D density of companies in the internal manifold coordinates. The second graph is the "3D–Principal Manifold Graph", it displays the companies in the first three principal component coordinates.

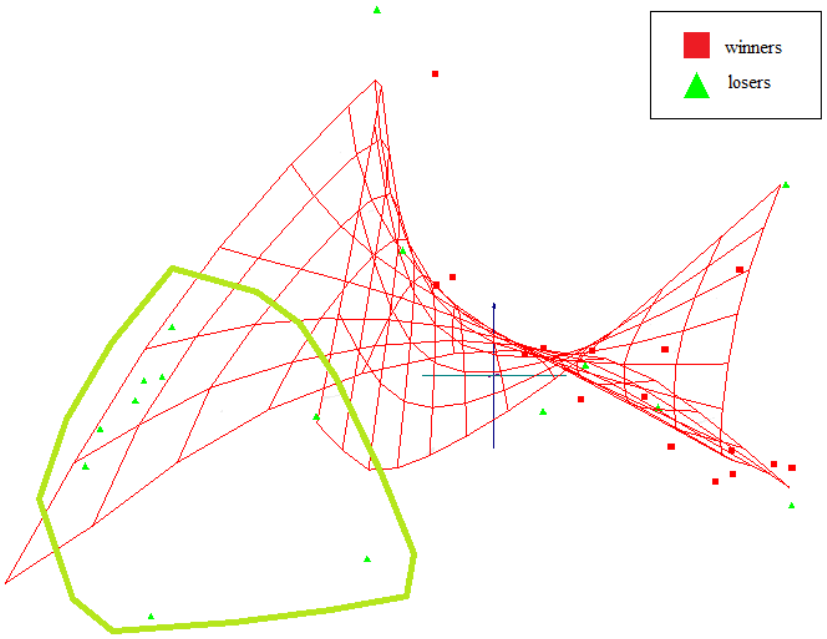
For the HANGSENG index with first 3–month time frame, Figure 2.10 of elastic maps are generated. From (a), it shows that the winners and losers are almost separated nicely. The losers generate 4 "clusters" within the internal coordinates. These "clusters" are located on each corner of the coordinate plane. From (b), it shows the winners are closer to each other than the loser companies as the red points are more compact to each other. However, the map boundary is not linear since some loser companies that are very close to the winner companies. There are around six points close to the red points. These green points can be considered as outliers. As the error of kNN is dominated by the structure of data, this visualization supports the LOOCV error analysis.

For the DAX index, Figure 2.11 shows elastic maps have a bit worse results than the one for the HANGSENG index. From (a), the losers generate 2 main "clusters" one near the top left corner and another one on the right–hand side. It is still possible to separate the "winner" and "loser" companies but it is not as clear as the result for the HANGSENG index. From (b), it seems that the "winner" company are close to each other in the middle of the map while the losers formed two "clusters" on the bottom. Both "winner" and "loser" companies have similar closeness between their own points. Hence from the figure, it is different to the kNN analysis that there is still a small amount of possibility of predicting the future "success".

For the elastic maps of companies of the FTSE index (Figure 2.12) and the NASDAQ index (Figure 2.13), there is no sign of separation. There is a small cluster of "loser" companies for the FTSE index but there are no clusters for "winner" companies. For the NASDAQ, there is no clear sign of clusters in the internal coordinate maps. The elastic maps give negative results for the companies of these

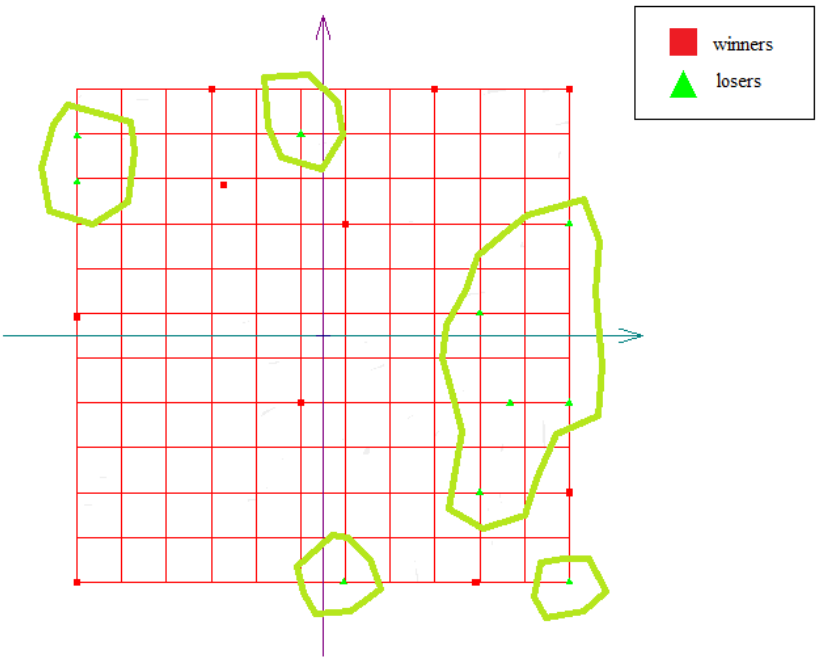


(A) 2D–Elastic Map

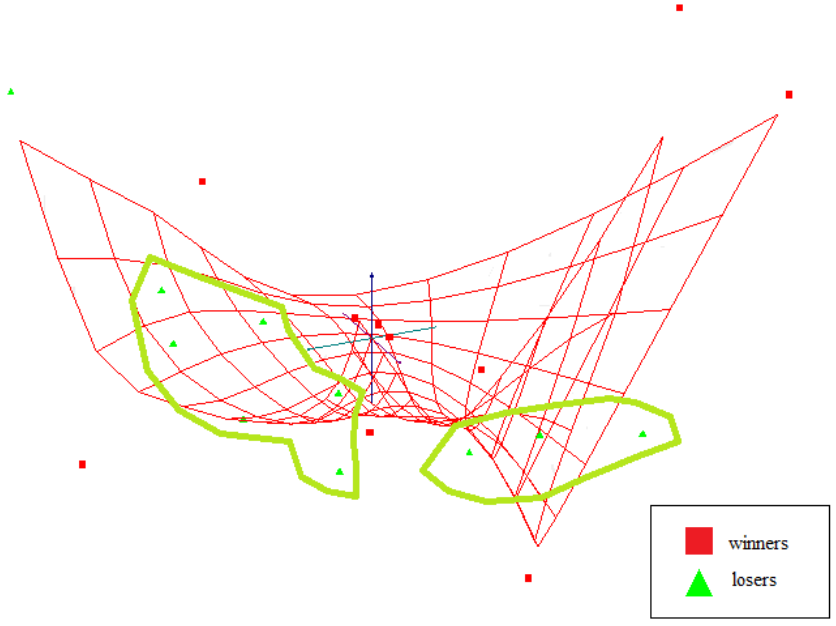


(B) 3D–Principal Manifold Graph

FIGURE 2.10: Visualization of components (companies) of HANGSENG index (log–returns) using elastic maps: (a)2D–Elastic Map (b)3D–Principal Manifold Graph The Hand–made green lines show the "clusters" of loser companies

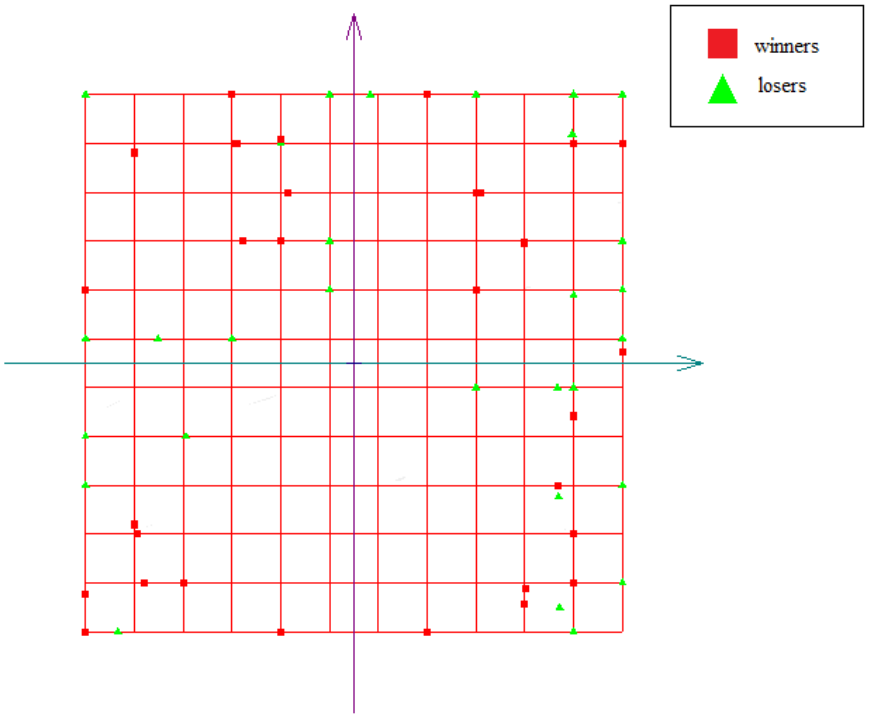


(A) 2D–Elastic Map

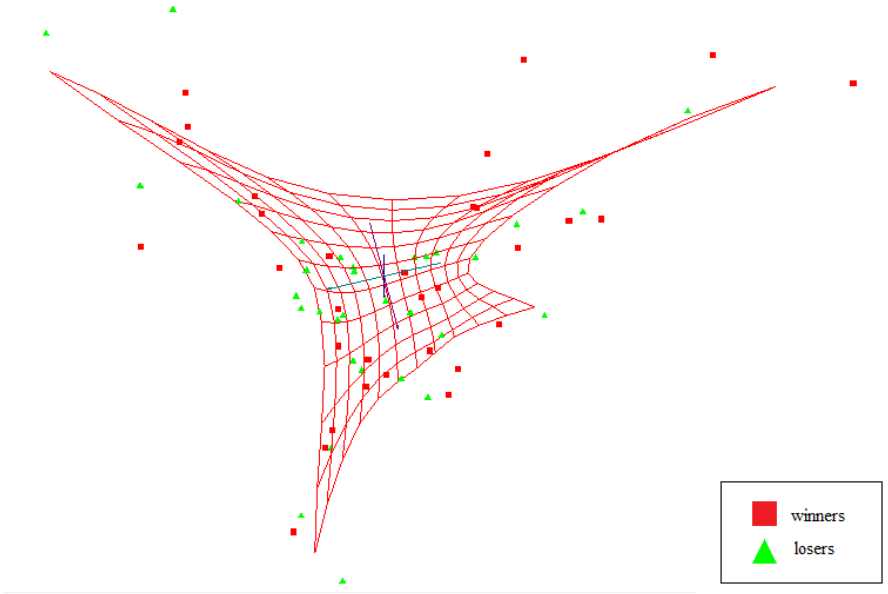


(B) 3D–Principal Manifold Graph

FIGURE 2.11: Visualization of components (companies) of DAX index (log–returns) using elastic maps: (a)2D–Elastic Map (b)3D–Principal Manifold Graph The Hand–made green lines show the "clusters" of loser companies

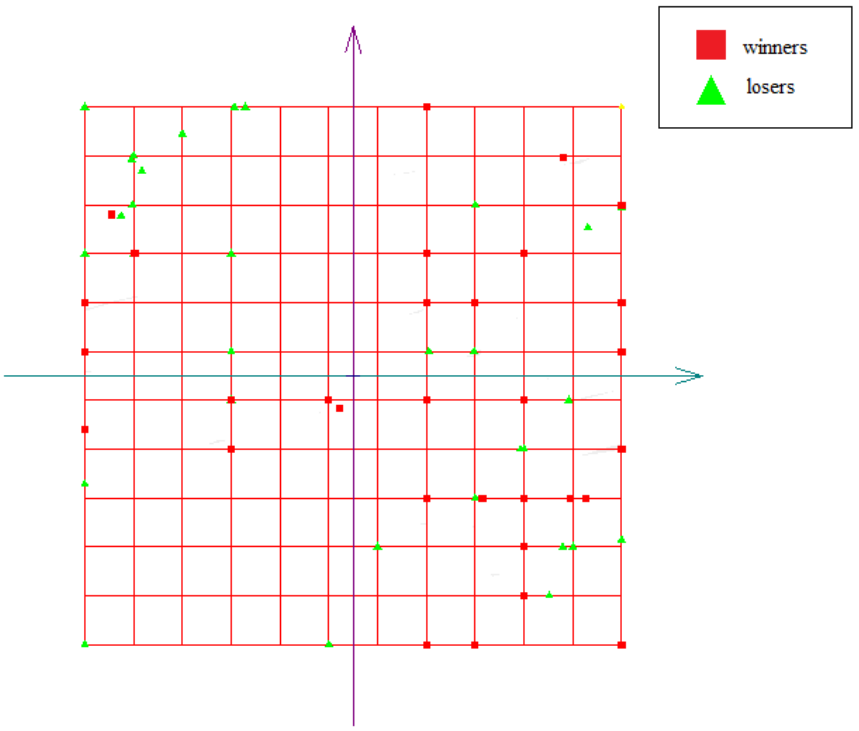


(A) 2D–Elastic Map

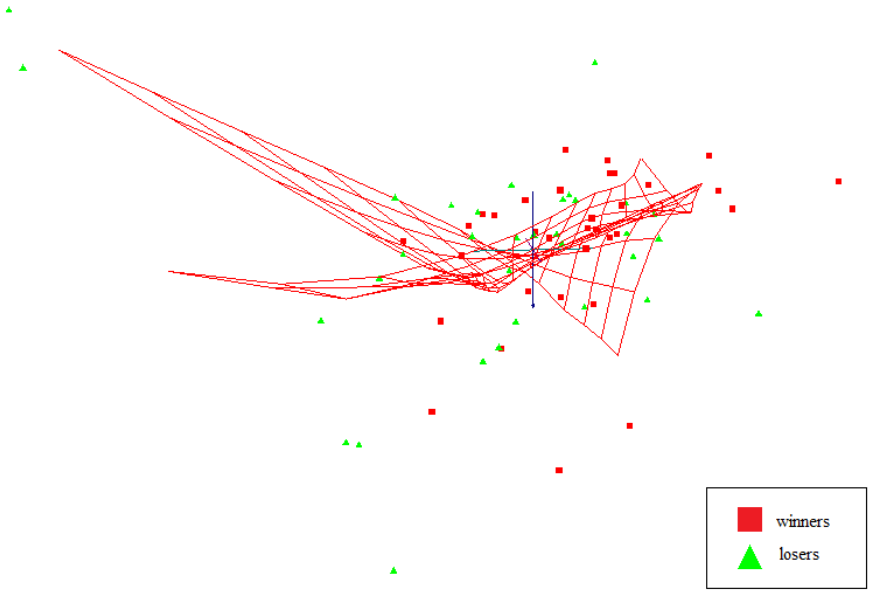


(B) 3D–Principal Manifold Graph

FIGURE 2.12: Visualization of components (companies) of the FTSE index (log–returns) using elastic maps: (a)2D–Elastic Map (b)3D–Principal Manifold Graph The Hand–made green lines show the "clusters" of loser companies



(A) 2D–Elastic Map



(B) 3D–Principal Manifold Graph

FIGURE 2.13: Visualization of components (companies) of the NASDAQ index (log–returns) using elastic maps: (a)2D–Elastic Map (b)3D–Principal Manifold Graph The Hand–made green lines show the "clusters" of loser companies

Winner (Distance)			Loser (Distance)		
	μ	σ		μ	σ
3 months	0.0625	0.0605	3 months	0.3750	0.1210
6 months	0.1875	0.0976	6 months	0.4375	0.1240
9 months	0.1875	0.0976	9 months	0.4375	0.1240
12 months	0.1875	0.0976	12 months	0.3750	0.1210
15 months	0.1875	0.0976	15 months	0.3750	0.1210
18 months	0.1875	0.0976	18 months	0.3750	0.1210
Winner (Proximity)			Loser (Proximity)		
	μ	σ		μ	σ
3 months	0.0625	0.0605	3 months	0.3750	0.1210
6 months	0.1875	0.0976	6 months	0.4375	0.1240
9 months	0.1250	0.0827	9 months	0.4375	0.1240
12 months	0.1250	0.0827	12 months	0.3750	0.1210
15 months	0.1250	0.0827	15 months	0.3750	0.1210
18 months	0.1875	0.0976	18 months	0.3750	0.1210

TABLE 2.2: Table of result using proportion estimate analysis for "winner" and "loser" companies using Distance and Proximity. μ is the sample distribution mean, and σ is the sample distribution standard deviation.

two indices. The kNN classifier is structure sensitive the elastic maps support the LOOCV error analysis. Therefore, the experiments have negative results and there is no evidence showing the possibility of predicting future success.

Consider the error to be the probability of occurrence event, the sample distribution of error is computed by applying proportion estimate. Table 2.2 shows the results of proportion estimate analysis. In general, the "loser" companies have sampling standard deviation of $0.1210 > 0.1$ where the sampling standard deviation for "winner" companies are approximately $0.0900 < 0.1$. The results using different correlation distances do not make much difference. There is no overlap in the interval between "winner" and "loser" companies.

2.4 Conclusions

In this case study, a backward analysis is used as the principal idea of experiments. The past and future data are used for labeling but only the past data is used in kNN classification. From the experiments, the results show that in this specific period of time, there is a phenomenon (i.e. the winner companies are closer to each other) for the HANGSENG index which means that there could be some conclusion about

the predictability of long–term success of companies in the HANGSENG index. Using different expressions of distances do not make much difference in the kNN LOOCV error analysis. For the only index with positive result, the more closing prices given (i.e. the longer initial time frame for initial information) does not improve the predictability for the kNN predictor since the LOOCV error analysis does not decrease when time period is larger.

For experiment results of the DAX index, the error analysis rejects the possibility of future prediction. However, the for the first 3–month experiment, the LOOCV error is just a bit below than 50%. Therefore, this is the boundary case for our experiment and it is the reason why it gives different results in the visualization result of the elastic map. In this case, it can be concluded that there is still a small amount of possibility of future success prediction. For the FTSE and the NASDAQ indices, it can be concluded that the long–term success is not predictable on the basis of daily closing prices in the initial time frame as the LOOCV error analysis is higher than 50%.

The positive result can also indicate for the HANGSENG index, the past prices have information about future prices. Hence, there is a high possibility that technical analysis is profitable for this case. Since the HANGSENG index is the index of developing market, this success in predictability can show that for the index of developing markets, the long–term success is more predictable. Technical analysis may be applied these young markets since for the HANGSENG and the DAX index, the historical prices may have some information. For the FTSE and the NASDAQ index, the prices have less information about the long–term success.

Chapter 3

A kNN historical Monte Carlo approach of modeling and predict daily stock returns

3.1 Introduction

Financial time series is one of the raw data of financial markets. It is a series of prices against time within a pre–defined time interval. Examples of financial time series are stock and share prices, commodity prices, currency prices etc. The stock returns are computed from the stock prices time series and they are commonly used in the study of predictability of stock returns. There is a significant degree of predictability in monthly stock returns in terms of the economic context [83]. In the short term horizon, the stock returns are predictable since there is a positive result in bivariate regression with short rates [84]. For a significant positive predictability, it is much easier to forecast the future. On the other hand, the predicted models would give a more accurate prediction.

Forecasting the prices of time series is widely studied but it is extremely difficult to obtain a good prediction result. Many forecast methods had been studied for years on these topics such as fundamental analysis, technical analysis (trend and patterns analysis), regression analysis, stochastic modeling etc. Stochastic modeling in time series forecasting could be defined as a process to analyze the probability distribution of the output of the model under assumptions that some

of the inputs are random. These random inputs are usually selected from the historical data from this time series.

Studies of technical analysis and efficient market hypothesis (EMH) leads to the question: do historical prices contain information on future prices? The motivation of this chapter is to use a kNN experiment constructed with the application of the dynamical system to look for k nearest neighbors of a single period closing price time series from a selection of historical prices time windows and use these nearest neighbors to construct predictions for the single period closing price time series. This approach is a universal method aimed for both ergodic dynamic system or random processes. Hence, this approach is applicable both in the predictable market (assumptions of technical analysis) and non–profitable market (assumptions of EMH). We use only the historical data to predict the future data and 95% confidence interval is used to analyze the predicted prices.

The experiment in this chapter, a new idea of stochastic modeling using kNN algorithm and application of dynamics systems is demonstrated. For a chosen market and a chosen time interval, the raw data are collected from the internet as closing prices. The daily log–returns are computed and for each component of the market index, "history", "present" and "future" part time fragments are defined. Then for "present" part of time fragment, the nearest neighbors time fragments are chosen from the "history" part of time fragments by application of kNN algorithm. The predicted "future" would be the next time fragment after the "present" part time fragment. This predicted "future" time fragment is then transformed from the log–returns into closing prices by using the idea of dynamic system and they are visualized to compare with the original closing prices.

This chapter is arranged as follows: Section 3.2 outlines the background knowledge, methodology and working hypothesis of this experiment; Section 3.3 demonstrate the result and analysis of different experiments (different parameters and different market). Section 3.4 demonstrate the comparison the ARMA(1,1) model prediction and kNN prediction. Section 3.6 conclude this chapter.

3.2 Backgrounds and methodology

3.2.1 The data selection and the data pre–processing

The raw data selected are the time series of daily closing prices of selected stock and share market index components. The closing prices are selected within a pre–defined time interval and they are downloaded from the Yahoo!Finance website. The closing prices are sorted from oldest to most recent. By comparing the length of components time series with the length of market index time series, the closing prices are considered as missing values if on the specific date when the market index has closing price while on the same date, the components time series do not have a closing price. For the case when the components time series have closing prices while the market index does not have the closing price on the specific date, this closing price of the component time series is deleted. The missing values are filled using linear interpolation hence the closing prices of components time series should have the exactly same length as the closing prices of market time series. The daily log–returns of the closing prices are then computed and they are used as the data for the kNN experiment. The daily log–returns time series for a component time series of length T is defined as:

$$LRts(i+1) = \ln\left(\frac{CPts(i+1)}{CPts(i)}\right) \tag{3.1}$$

where $i = 1, 2, \ldots, T-1$ and $CPts(i)$ is the closing prices of this component at date i. Use of daily log–return rather than plain closing prices has several advantages. The daily log–return have mean value around zero and some good statistical properties. Since we are interesting in searching for the nearest neighbors, it has a greater possibility to found the similar log–returns rather than the closing prices. Note that the log–return time series only contains one less element than the closing prices time series. (i.e. for a component time series of length $T-1$, the length of log–returns time series is $T-1$.)

The market index measures the prices movements for components of the market index. The closing prices of the market index are generally computed from the closing prices of components, they indicate the general trend of the market prices and help investors to make trading decisions. The market index could be defined by different regions of markets. The Deutscher Aktien index (or DAX index) was opened on the 1st October 1988. It is the blue chip stock and shares index of

Name	Category	Name	Category
ADS	Clothing	ALV	Insurance
BAS	Chemicals	BAYN	Pharmaceuticals and Chemicals
BEI	Consumer goods	BMW	Manufacturing
CBK	Banking	CON	Manufacturing
DAI	Manufacturing	DB1	Securities
DBK	Banking	DPW	Communications
DTE	Communications	EOAN	Energy
FME	Medical	FRE	Medical
HEI	Building	HEN3	Consumer goods
IFX	Manufacturing	LHA	Transport Aviation
LIN	Industrial gases	LXS	Chemicals
MRK	Pharmaceuticals	MUV2	Insurance
RWE	Energy	SAP	IT
SDF	Chemicals	SIE	Industrial, electronics
TKA	Industrial, manufacturing	VOW3	Manufacturing

TABLE 3.1: Table of names and categories of all DAX components from Wikipedia website.

the German market. The DAX index measures the prices movements of 30 major components traded in Frankfurt Stock Exchange.

Table 3.1 lists the names and categories for all 30 components of DAX index. For experiment described later in this paper, the closing prices of DAX components are downloaded from Yahoo!Finance website within the time interval from 04–Jan–2010 to 30–Dec–2011. Approximately one year of closing prices are downloaded from the website and after data–preprocessing step, the length of closing prices time series for each component is 516. Figure 3.1 is the graph of DAX index closing prices from 04–Jan–2010 to 30–Dec–2011. It is observed that the price drop from beginning to approximately date 70. Then it bounced up and down around 6000 EUR until at date 200, a general upward trend is formed. At date 300 there is a small drop in the price but it goes up again when there is another strong decreasing trend at date 400. The prices reach 5100 EUR at around date 430 and it starts to oscillate around 6000 EUR again till the end of the time interval. Hence, it could tell within this interval, it shows the time moment before the market collapsed at around date 400. It is interesting to see if the history could have any information on this collapsed market.

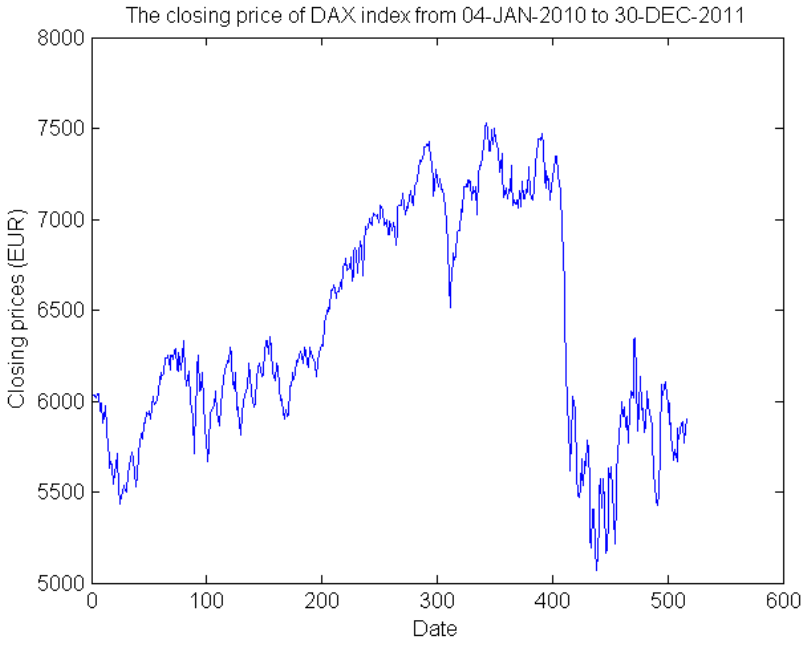


FIGURE 3.1: Graph of closing prices for DAX index from 04–Jan–2010 to 30–Dec–2011

The components are defined into different categories by the nature of their business. The financial sectors could contain categories such as Banks, Insurance, Investment, Securities, financial series etc. It is interesting to see the performance of components time series for this sector. For DAX index, there are 5 components in the financial sector (i.e. components ALV, CBK, DB1, DBK, and MUV2).

The Financial Times Stock Exchange 100 Index (or FTSE100 index) was opened on 3rd January 1984. This index is held by FTSE group and it measures the top 100 highest capitalization components of the market in the UK. The FTSE100 is traded in London Stock and Exchange and it contains 100 components. It is also taken approximately 81% of capitalization traded in London Stock and Exchange. The list of components of FTSE100 components from Wikipedia website contains 101 components since component RSD has two shares, that is A class and B class. Table 3.2 is the list of financial sector components of FTSE100 index. There are 18 components in this sector with categories such as Banks, Financial Services, Life/Nonlife Insurance. The closing prices are selected in a time interval from 02–Jan–2012 to 31–Dec–2013 for approximately 1 year. Originally there were 19 components in this sector but one of them was deleted from this list since for this component the length of time series is too short (too many missing values). After the raw data is downloaded from the website and the time series of each component after data pre–processing step has length of 522. Figure 3.2 shows the

Name	Category	Name	Category
ADM	Nonlife Insurance	ADN	Financial Services
AV	Life Insurance	BARC	Banks
HSBA	Banks	HL	Financial Services
III	Financial Services	LGEN	Life Insurance
LLOY	Banks	LSE	Financial Services
OML	Life Insurance	PRU	Life Insurance
RBS	Banks	RSA	Nonlife Insurance
SDR	Financial Services	SL	Life Insurance
STAN	Banks	STJ	Life Insurance

TABLE 3.2: Table of names and categories of selected FTSE100 components from London Stock Exchange website for financial sectors.

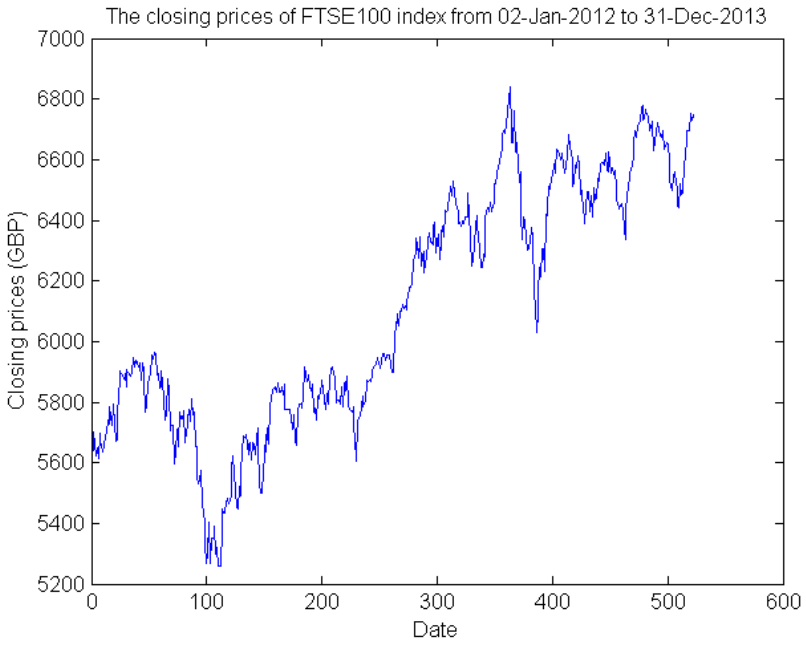
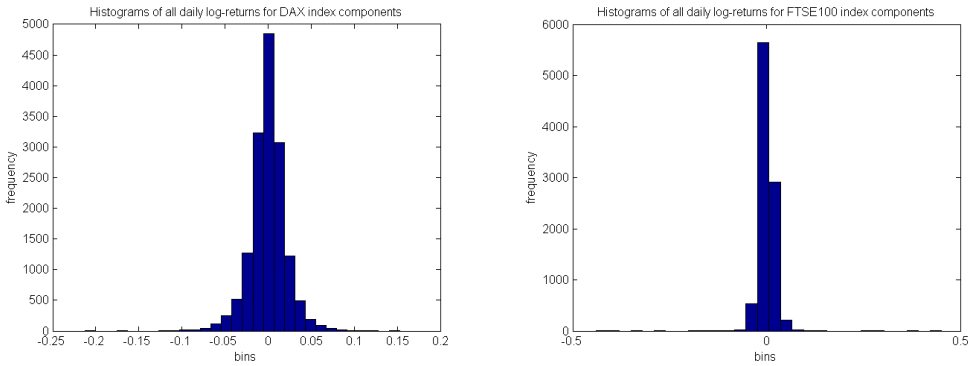


FIGURE 3.2: Graph of closing prices for FTSE100 index from 02–Jan–2012 to 31–Dec–2013

closing prices of FTSE100 index from 02–Jan–2012 and 31–Dec–2013. The prices starts with around 5700 GBP in the beginning, then it goes up to 6000 GBP at around date 65. Then it decreases and reach 5300 GBP at around date 100. The prices then have a general upward trend and reach around 7900 at date 390. In the end, the prices bounced up and down around 6600 GBP and in the end the price is approximately 6700. The FTSE100 index has a general increasing trend within this time interval, hence it could conclude the market capitalization is rising during this time interval.

From Figure 3.3 it is observed that the distributions of log–returns for DAX and



(A) DAX index (B) FTSE100 Index

FIGURE 3.3: Graphs of Histograms of log–returns for all components for DAX and FTSE100 index.

Index	DAX	FTSE100
Mean	-2.87×10^{-5}	1.2×10^{-3}
Medean	0	5.68×10^{-4}
Mode	0	0
Std.	2.05×10^{-2}	2.06×10^{-2}
Kurtosis	7.04	1.11×10^{2}
Skewness	-1.75×10^{-1}	-1.29

TABLE 3.3: Table of descriptive statistics for log–returns of all components for DAX and FTSE100 index.

FTSE100 index forms bell shape distribution. For both indices, the distributions are symmetrical around 0 but the log–returns of DAX index components has a much lower kurtosis than the log–returns of FTSE100 index components as seen from Table 3.3. The peak of histogram for FTSE100 index is higher than the peak of histogram for DAX index.

3.2.2 The kNN algorithm and methodology

After the data pre–processing step, the data is partitioned into three parts, the "history", "present" and "future". The key idea of experiments is to find the nearest neighbor log–return time fragment for the "present" part from the "history" part time fragments. The closing price time series for "future" part can be regenerated by applying a function of the nearest neighbor log–return time fragment. This regeneration of future prices is an application of Taken's embedding theorem on time series. The generalized idea is studied as the nearest trajectory strategy and it shows that the dynamical system could be reconstructed by lagged time windows accurately [85] [86]. Predictions of time series would have two common

forms, one–steps prediction and free–run prediction [87]. The method described here would be an example of a free–run prediction since the regeneration of the "future" prices would need a subset of time–lagged log–returns. By analyzing the predicted prices using kNN, it can conclude if this prediction method works for both ergodic time series and stationary time series.

The kNN algorithm is one of the most popular algorithms in machine learning. It is a model–free algorithm and requires no training process. It is widely used in classification and regression analysis. The output of this algorithm is the set of k points that are the "closest" points to the query point. Let us define the query point to be X_{query} and the set of points that the algorithm selects nearest neighbors from being $Y = \{Y_1, Y_2, \ldots, Y_n\}$, where there are n points in this set. All points have dimension p. The measure of closeness is computed between the query point and every point in the set Y such that $d(i) = f(X, Y_i)$ where $f()$ is a defined function depending on the type of distance metric used in the algorithm. Then the k points in Y contains the least distance are considered as the k nearest neighbors. In our experiment, the time fragment of each component in "present" part is considered as the query point. The time fragments of "history" part are the set Y.

Before applying the kNN algorithm, the length of a time fragment need to be defined. A time fragment of the time series is part of whole time series with fixed length. Let us define τ_{pre} be the length of time fragment of "present" part, τ_{fut} be the length of time fragment of "future" part and t be the date of time series for each component, such that $t = \{1, 2, \ldots, T\}$ and $t = 1$ represents date 1 and date T represents the last date of the whole price time series. Let us define the date of "history" part be $t_{\text{history}} = \{2, 3, \ldots, T - \tau_{\text{pre}} - \tau_{\text{fut}}\}$, the date of "present" part be $t_{\text{present}} = \{T - \tau_{\text{pre}} - \tau_{\text{fut}} + 1, \ldots, T - \tau_{\text{fut}}\}$ and the date of "future" part be $t_{\text{future}} = \{T - \tau_{\text{fut}} + 1, \ldots, T\}$.

The nearest neighbors log–return time fragment of each component in "present" part are selected from "history" part. The "sliding window" strategy is used for getting the time fragment from the whole "history" part log–return time series. For a "history" part time series $\{LRts(2), LRts(3), \ldots, LRts(T_{\text{hist}})\}$ of length T_{hist}, let us define "window" of the log–return time series be a subset time series with fixed length τ_{pre}. We start to define the "window" be $\{LRts(2), LRts(3), \ldots, LRts(\tau_{\text{pre}} + 1)\}$ and let the first time fragment be this "window". Then the "window" is

shifted to the next date, therefore, the next "window" is defined as $\{LRts(3), LRts(4), \ldots, LRts(\tau_{\text{pre}} + 2)\}$. The next time moment is assigned to this "window". This process would repeat until it the last time fragment is defined as $\{LRts(T_{\text{hist}} - \tau_{\text{pre}} + 1), LRts(T_{\text{hist}} - \tau_{\text{pre}} + 2), \ldots, LRts(\tau_{\text{pre}})\}$. The general formula of the time fragment for this "history" part can be defined as:

$$Frag(i) = \{LRts(1 + i), LRts(2 + i), \ldots, LRts(1 + \tau_{\text{pre}} + i)\} \tag{3.2}$$

where $i = \{1, 2, \ldots, T_{\text{hist}} - \tau_{\text{pre}} + 1\}$.

The experiment may vary on different formations of "history" part. There are 3 parameters in the experiments, the number of nearest neighbors k, the length of "present" part τ_{pre} and the length of "future" part τ_{fut}. For DAX index, the whole experiment uses all historical prices across all components as the "history" part. This experiment uses a relatively large size of "history" part. For the partition of the log–returns, the "history" part is defined as the collection of log–returns from date 2 to date 297, the "present" part is defined as the collection of log–returns from date 298 to date 357, the "future" part is defined as the collection of log–returns from date 358 to date $357 + \tau_{fut}$ for different values of τ_{fut}. If we have $k = 60$, $\tau_{\text{pre}} = 60$ and $\tau_{\text{fut}} = 40$, and the length of each component in "history" part is 296. By applying the "sliding window" strategy, the "history" part contains $(296 - 40) \times 30 = 7680$ time fragments. The number of time fragments does not reach the end to avoid the case when there is a overlap between the set of predicted "future" part log–returns and the set of "present" part log–returns. This small process ensures that the predicted "future" part log–returns are from the "history" part. For the whole experiment of DAX index, changing of τ_{fut} means the change in the length of the "future" part. This experiment also uses $\tau_{\text{fut}} = 60$ and $\tau_{\text{fut}} = 100$ in order to see the performance of experiments for a longer future.

Different formation of "history" part could lead to different results. The independent experiment of DAX index searches for a specific component nearest neighbor log–returns time fragments from "history" of this component itself. Therefore, for a specific component, the "history" part is much smaller compared with the "history" part of DAX index whole experiment. In this experiment we set $k = 60$, $\tau_{\text{pre}} = 60$ and $\tau_{\text{fut}} = 40$. There are 256 fragments computed from "history" part for each component. The financial sector experiment of DAX index is slightly different. Only the components of the category in finance are used as the raw data.

The "history" part contains the components of financial sector only. Hence there are $256 \times 5 = 1280$ time fragments computed from the "history" part.

3.2.3 Various Distance Measurement

The distance measurements are key factors in kNN algorithm. Different metrics could lead to different results and it is interesting to see the results of different distance metrics and similarities. In our experiments three distance metrics (i.e. Euclidean distance, city block distance and correlation distance) and one similarity (i.e. cosine similarity) are applied. The Euclidean distance or Euclidean norm between points $\mathbf{p}$ and $\mathbf{q}$ such that $\mathbf{p} = (p_1, p_2, \ldots, p_n)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_n)$ is computed as:

$$d_{Euc}(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{3.3}$$

if $\mathbf{p}$ and $\mathbf{q}$ are in n–dimensional space. This is the measurement of length between two points in Euclidean space. Another metric used is the City Block distance. The City Block distance or Manhattan distance between $\mathbf{p}$ and $\mathbf{q}$ is defined as:

$$d_{Cit}(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^{n} |p_i - q_i| \tag{3.4}$$

and this is widely used in urban design. The correlation distance uses concept of product moment correlation coefficient. It is computed as:

$$d_{Cor}(\mathbf{p}, \mathbf{q}) = 1 - \frac{\sum_{i=1}^{n} (p_i - \bar{p})(q_i - \bar{q})}{\sqrt{\sum_{i=1}^{n} (p_i - \bar{p})^2} \sqrt{\sum_{i=1}^{n} (q_i - \bar{q})^2}} \tag{3.5}$$

where $\bar{p} = \frac{1}{n} \sum_{i=1}^{n} p_i$ and $\bar{q} = \frac{1}{n} \sum_{i=1}^{n} q_i$. One similarity is applied in this experiment. Unlike distance matrix, the similarity measure has large values on similar objects and negative or zero value for objects that are very dissimilar to each other. The cosine similarity measures the cosine of the angle between the two points. This is computed as:

$$d_{Cos}(\mathbf{p}, \mathbf{q}) = 1 - \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} {p_i}^2} \sqrt{\sum_{i=1}^{n} {q_i}^2}}. \tag{3.6}$$

These distance metrics would select different nearest neighbors for the same point. Different prediction results of selecting nearest neighbors would be compared in the later sections to analyze their performance on this method.

3.2.4 The Taken's theorem and regeneration of predicted "future" part time series

Taken's theorem shows that under several states the dynamics system could be reconstructed from sequences of dynamic systems [88]. This theorem could also be applied in forced and stochastic systems and the theorem could have slightly different forms [89]. Let us define a dynamical system with state $x(t) \in \mathbb{R}$ which is solution of some differential equation:

$$\dot{x} = \psi(x) \tag{3.7}$$

where ψ is a smooth function and $\psi : \mathbb{R}^N \to \mathbb{R}^N$. For a submanifold $\mathcal{M} \subset \mathbb{R}^N$, let us define a flow function $G : \mathcal{M} \times \mathbb{R} \to \mathcal{M}$ with state $x(t) \subset \mathcal{M}$ by:

$$x(t_0 + T) = G_T(x(t_0), T), \tag{3.8}$$

for some real number T and:

$$x(t_0 + kT) = G_T^k(x(t_0), T) = \underbrace{G_T \circ G_T \circ \cdots \circ G_T}_{\text{k times}}(x(t_0)) \tag{3.9}$$

for some positive integer k. For a smooth measurement function ϕ where $y(t) = \phi(x(t))$ such that $\phi = \mathbb{R}^N \to \mathbb{R}$. A delay map with M delays $F_{(\phi, G_{-T_s}):\mathcal{M}\to\mathbb{R}^M}$ is defined as:

$$\begin{aligned} F(x(t) &= F_{(\phi,-G_{T_s})}(x(t)) \\ &= [y(t), y(t-T_s), \cdots, y(t-(M-1)T_s)]^T \\ &= [\phi(x(t)), \phi \circ G_{-T_s}(x(t)), \cdots, \phi \circ G_{-T_s}^{M-1}(x(t))]^T, \end{aligned} \tag{3.10}$$

for sampling time T_s. A version of Taken's embedding theorem is stated as [90]:

Theorem 1. *Let $\mathcal{M}$ be a compact manifold of dimension K and suppose we have a dynamical system defined by 3.7 that is confined on this manifold. Let $M > 2K$ and suppose:*

1. *the periodic points of G_{-T_s} s with periods less than or equal to $2K$ are finite in number and,*
2. *G_{-T_s} has distinct eigenvalues on any such periodic points.*

Then the observation functions ϕ for which the delay coordinate map F 3.10 is an embedding form an open and dense subset of $\mathcal{C}^2(\mathcal{M}, \mathbb{R})$.

The idea of Taken's theorem is applied in the current paper to regenerate the closing prices of "future" part time series from the "future" part log–return time series. After k nearest neighbors are selected, the log–return time fragment are selected by selecting the log–returns of next τ_{fut} dates after the last log–return prices of the nearest neighbors. Then the system of closing prices time series are regenerated from these log–return time series. Let us define the last closing price of the "present" part of log–return time series to be defined as $ts^{\text{com}}_{\text{last}}$. The nearest neighbor time series is defined as $NNts^{\text{com}}(j)$ where $j = 1, 2, \ldots, \tau_{\text{fut}}$. The regenerated time series could be defined as a flow map using the inverse function of 3.1:

$$\begin{aligned} ts^{\text{com}}_{\text{fut}}(j) &= ts^{\text{com}}_{\text{fut}}(j-1)e^{futLR^{\text{com}}(1)} \\ &= ts^{\text{com}}_{\text{fut}}(j-2)e^{futLR^{\text{com}}(2)} \times e^{futLR^{\text{com}}(1)} \\ &= ts^{\text{com}}_{\text{last}}e^{futLR^{\text{com}}(j)} \times e^{futLR^{\text{com}}(j-1)} \times, \ldots, \times e^{futLR^{\text{com}}(1)} \end{aligned} \tag{3.11}$$

where $futLR^{\text{com}}(j)$ represent the log–return time series for "future" part for jth date within the date of "future", $j = 1, 2, \ldots, \tau_{\text{fut}}$ and $ts^{\text{com}}_{\text{fut}}(0) = ts_{\text{last}}$.

3.2.5 Visualization of experiment result

The regeneration of k predicted prices of "future" part would produce k time series. To analyze the predicted, the plot of 95% confidence interval for each component is plotted together with the real prices for "present" part and "future" part. The 95% confidence interval plot with average of predicted prices is computed if $k > 1$, i.e. $\mu \pm 1.96\sigma$ where $\mu = \frac{1}{k}\sum_{i=1}^{k} futts(i)$ and

$$\sigma = \sqrt{\frac{1}{k-1}\sum_{i=1}^{k}(futts(i) - \mu)^2}. \tag{3.12}$$

The 95% confidence interval computes the range of values that means a good estimation of real prices. If the real prices lies inside this interval, then this would indicate that this method is giving a stochastically prediction. Together with the real price in the future, it is straight forward to compare with the original price for "future" part.

3.2.6 The development of the GARCH model

In 1983, the autoregressive conditional heteroscedastic (ARCH) processes were introduced in field of econometrics. Rather than assuming forecast variance to be constant, these processes are uncorrelated processes with zero mean and non–constant conditional variances on the past values but constant unconditional variances [91].

Definition 3.1. The discrete time stochastic processes $\{\varepsilon_t\}$ are referred as an ARCH model with the form:

$$\begin{aligned} &\varepsilon_t = z_t\sigma_t, \\ &z_t \text{ i.i.d.}, \mathrm{E}(z_t) = 0, \mathrm{var}(z_t) = 1, \end{aligned} \tag{3.13}$$

where σ_t is positive and depends on time. σ_t is modeled as:

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2,$$

where $\alpha_0 > 0$, $\alpha_i \geq 0$, and $i > 0$.

ε_t can be both a univariate process and multivariate process. The ARCH modelling in finance is reviewd and the extensions of ARCH modelling such as the GARCH model, the EGARCH model etc [92]. The linear ARCH(q) model is suggested as a linear function of past squared values of the process.

Definition 3.2. Consider a process for σ_t, the linear ARCH(q) model compute σ_t^2 as linear function of past squared values of the process,

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2,$$

where $\omega > 0$ and $\alpha_i \geq 0$.

ARCH model can be applied in estimating the variance for U.S. inflation [91] and United Kingdom inflation [93]. This model is applied in Value–At–Risk (VAR) model [94]. Combining linear ARCH model with moving average model, the linear generalized ARCH(p) model or the GARCH(p, q) model is developed.

Definition 3.3. The linear Generalized ARCH(p) or GARCH(p, q) model [95] was introduced as:

$$\sigma_t^2 = \omega + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2,$$

where σ_t^2 is nonnegative.

The GARCH(p, q) model is applied in estimating the conditional variances of asset [96] [97]. Consider innovation z_t has Gaussian or student's t distribution with $\nu > 2$. If the history of a process h_t at time $t = 1, 2, \ldots, N$ is given and the innovations are conditionally independent. The likelihood function for the innovation is given as:

$$f(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N | h_{N-1}) \prod_{t=1}^{N} f(\varepsilon_t | h_{t-1}),$$

where f is a standardized Gaussian or t density function. If z_t has a standard Gaussian distribution, the likelihood function is:

$$-\frac{N}{2}\log(2\pi) - \frac{1}{2}\sum_{t=1}^{N}\log\sigma_t^2 - \frac{1}{2}\sum_{t=1}^{N}\frac{\varepsilon_t^2}{\sigma_t^2},$$

if z_t has a standardized student's t distribution with $\nu > 2$ degrees of freedom, the likelihood function is:

$$N\log\left(\frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\pi(\nu-2)}\Gamma\frac{\Gamma}{2}}\right) - \frac{1}{2}\sum_{t=1}^{N}\log\sigma_t^2 - \frac{\nu+1}{2}\sum_{t=1}^{N}\log\left(1 + \frac{\varepsilon_t^2}{\varepsilon_t^2(\nu-2)}\right).$$

After p, and q is chosen, parameters such as α_i and β_i can be estimated via maximum likelihood method. The stock returns volatility in Tokyo Stock Exchange from 1986 to 1989 is forecasted using the GARCH models [98]. The volatility of FTSE–100 index is forecasted using the GARCH model [99] and it is shown that future trading increases volatility. A pricing method of Hang Seng index options around the Asian financial crisis using the GARCH model [100]. By comparing with the Black–Scholes model, the GARCH approach have good performance.

The comparison between the kNN experiment and the GARCH model is interesting but the kNN experiment predict stock daily returns while the GARCH model predicts volatility and conditional variance with given historical prices. By using idea of integrated model, μ generated from the ARMA process and σ generated from the GARCH model [101], the next day return can be regenerated as:

$$\begin{aligned} R_t &= \mu + \sigma \times \text{i.i.d.} N(0,1) \\ X_t &= R_t + X_{t-1}. \end{aligned} \tag{3.14}$$

However, this approach does not work for log–returns hence the comparison between our the kNN experiment is incompatible. The ARMA model is used instead to predict the daily log–return and compared with the kNN experiment results.

3.2.7 The Autoregressive Moving Average model

In time series analysis, Autoregressive–moving–average model (or the ARMA model) is used to predict future time series with given history time series. This model is generalized in 1951 [102]. The ARMA(p, q) model or process is formulated as in [103]:

$$\tilde{z_t} = \phi_1 \tilde{z}_{t-1} + \ldots + \phi_p \tilde{z}_{t-p} + a_t - \theta_1 a_{t-1} - \ldots - \theta_q a_{t-q}. \tag{3.15}$$

In the polynomial form:

$$(1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p)\tilde{z}_t = (1 - \theta_1 B - \theta_2 B^2 - \ldots - \theta_q B^q)a_t \tag{3.16}$$

or, $\phi(B)\tilde{z}_t = \theta(B)a_t$ where $\phi(B)$ and $\theta(B)$ are polynomials with degree p and q in B. The ARMA(p, q) process could forecast time series with finite known history time series [104]. The parameters of the ARMA(p, q) process are computed by maximum likelihood estimation based on historical time series, then the future time series is fitted into a linear model with these parameters [105] [106]. The ARMA(p, q) process is widely studied and applied in studies of time series, it is one of the classical method of prediction. The comparison of the ARMA(1,1) model and the kNN experiment is shown in Section 3.4. For forecasting of closing prices using the ARMA(1,1) model, "history" part and "present" part are used as historical time series to predict "future" part time series. By comparing the forecasting results, conclusions on predictability of historical time series can be shown.

3.2.8 The Lorenz system

The Lorenz system is proposed by Edward Lorenz in 1973 [107]. It is a nonlinear deterministic system (a system with no randomness involved in generating its

future state) of three differential equations. The Lorenz system is defined as:

$$\begin{aligned} \frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma(y-x), \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= x(\rho-z)-y, \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy-\beta z, \end{aligned} \tag{3.17}$$

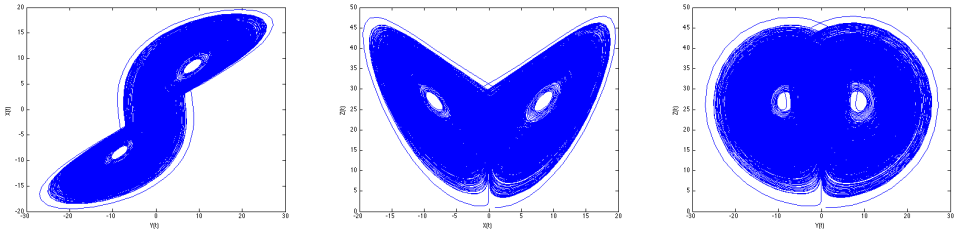
where t is time and the system parameters are σ, ρ and β. This solution has chaotic behaviour. Some elementary properties of the Lorenz system had been studied. The three parameters of the Lorenz system are assumed to be positive.

Proposition 3.4. *Suppose $\rho < 1$. Then all solutions of the Lorenz system tend to the equilibrium point at the origin.*

Proposition 3.5. *The equilibrium points $Q_{\pm}$ are sinks provided:*

$$1 < \rho < \rho^* = \sigma\left(\frac{\sigma+\beta+3}{\sigma-\beta-1}\right),$$

where the Hopf bifurcation occurs at ρ^* [108]. In the Lorenz attractor, Lorenz used $\sigma = 10$, $\beta = 83$ and $\rho = 28$. Figure 3.4 represents the plot of solution to



(A) $X(t)$ against $Y(t)$ (B) $Z(t)$ against $X(t)$ (C) $Z(t)$ against $Y(t)$

FIGURE 3.4: Example of Lorenz attractor

the Lorenz system. They are generated using MATLAB for $t = [1, 1000]$ and the initial condition is set to be $x(0) = 1$, $y(0) = 1$ and $z(0) = 1$ for simplicity.

3.2.8.1 The Lorenz attractor in kNN experiment

The Lorenz system is a deterministic system. Theoretically, if the initial condition is given, the future is achievable. In our kNN, we add a random noise to the Lorenz attractor with random initial conditions. Then we apply the kNN experiment on these data. MATLAB is used to compute the solution of system defined in (3.17). The algorithm of the the kNN experiment using the Lorenz attractor is described

Algorithm 6 The kNN experiment for data generated from the Lorenz attractor with noise

1. Set a time step for solution of differential system, i.e. $\tau = 0.5$.
2. Generate random initial conditions $x(0)$, $y(0)$, $z(0)$ using MATLAB function 'rand' (i.e. 'rand' function generates random numbers from $\mathrm{U}(0, 1)$).
3. Use MATLAB function 'ode45' to compute solution of the Lorenz system $x(t)$, $y(t)$ and $z(t)$ with σ, β and ρ from the Lorenz attractor for time $t \in [0, 1000]$ with time step size τ.
4. Choose one solution to the system, i.e. select $x(\Delta t)$ for $\Delta t = 1, 2, ..., 2000$. Set a noise level α_ϵ, i.e. $\alpha_\epsilon = 0$ and the noise is $\epsilon(t) = \alpha_\epsilon \times D(t)$ where $D(1), D(2), \ldots, D(t)$ are random samples from $\mathrm{U}(0, 1)$. Add noise to this solution, i.e. $x_\epsilon(\Delta t) = x(\Delta t) + \epsilon(t)$ for $\Delta t = 1, 2, ..., 2000$.
5. Define the first 1000 data points of $x_\epsilon(\Delta t)$ as 'history' part and the rest 1000 as test set.
6. Define a frame length for sliding window strategy, i.e. $n_{\text{window}} = 20$. In sliding window strategy, let 20 data points be 'present' and the next 20 be 'future'.
7. For each fragment of 20 data points in test set, select k nearest neighbors in collection of fragments of 'history' part. Use the next 20 data points of nearest neighbor as predicted 'future'.
8. Repeat Step 4 – Step 7 for $y(t)$ and $z(t)$.

in Algorithm 6. In this algorithm, sliding window strategy is applied for creating both set of fragments of both 'history' part and test set. Similarly to the other kNN experiment, the same sliding window strategy is used. The 'present' fragment length and 'future' fragment length are defined as 20 (or the 'window' length is set to 20). Hence, there are 960 'windows' or fragments in 'history' part and another 960 'windows' in test set. For each fragment in 'present' part, the k nearest neighbors are searched from set of 'history' part fragments. The predicted 'future' part is the next fragment of k nearest neighbors fragments. To analyze the accuracy of this experiment, mean squared error and standard deviation are used. From the Algorithm 6, for each 'window', there are k fragments. Consider $x^j_{\text{est}}(i)$ be predicted ith 'future' for 'window' j and $x^j(i)$ be real ith 'future' for 'window' j, where $m = 1, 2, \ldots, n_{\text{window}}$ and $j = 1, 2, \ldots, 960$. The mean squared error (MSE) for window j is defined as:

$$\mathrm{MSE}(j) = \frac{1}{n_{\text{window}}} \sum_{m=1}^{n_{\text{window}}} (x^j_{\text{est}}(m) - x^j(m))^2.$$

MSE measures accuracy of prediction for each nearest neighbor prediction for each 'window'. The average MSE for 'window' j is the average MSE of k nearest neighbor predictions for 'window' j (i.e. $\mathrm{MSE}_{\text{average}}(j) = \frac{1}{k} \sum_{i=1}^{k} \mathrm{MSE}^j(i)$). For predicted 'future' of each solution, there are 960 average MSE (i.e. one MSE

for each 'window'). Relative mean squared error (RMSE) is rather applied in the experiment to measure the relative difference between the estimated and real data. The relative mean squared error is defined as:

$$\mathrm{RMSE}(j) = \frac{\mathrm{MSE}(j)}{\frac{1}{n_{\mathrm{window}}}\sum_{m=1}^{n_{\mathrm{window}}}(x^j(m))^2},$$

where $\mathrm{RMSE}(j) \in [0,1]$. And the average RMSE is defined similarly as MSE (i.e. $\mathrm{RMSE}_{\mathrm{average}}(j) = \frac{1}{k}\mathrm{RMSE}^j(i)$). Another measurement applied in error analysis is the standard deviation. Consider we have k nearest neighbor predictions for each 'window' and let $i = 1, 2, \ldots, k$. The standard deviations of the ith nearest neighbor prediction in 'window' j are defined as:

$$\sigma^i_{k\mathrm{NN}}(m) = \sqrt{\frac{1}{960}\sum_{j=1}^{960}(x^m_{\mathrm{est}}(j) - \mu^m_{k\mathrm{NN}})^2},$$

where $m = 1, 2, \ldots, n_{\mathrm{window}}$ and $\mu^m_{k\mathrm{NN}} = \frac{1}{960}\sum_{j=1}^{960} x^m_{\mathrm{est}}(i)$. Hence, there are n_{window} standard deviation for each ith nearest neighbor prediction for each 'window'. The average standard deviation of k nearest neighbor prediction is the average of k standard deviations (i.e. average $\sigma_{k\mathrm{NN}}(m) = \frac{1}{k}\sum_{i=1}^{k}\sigma^i_{k\mathrm{NN}}(m)$).

3.3 Results and Analysis

Various experiments are performed in Section 3.3. The results are presented by the variance of log–return factors and 95% confidence intervals. It is also interesting to see the change in the results when the parameters and "history" part changes. For the DAX index, the whole experiment, independent experiment, and financial sectors experiment are performed. For FTSE100 index, only the financial sector experiment is performed. In different experiments, different formation for the set used to search for k nearest neighbors are formulated:

1. For the whole experiment, this set is a combination of time series trajectories of all components of the index in "history" part.

2. For the individual experiment, this set is a combination of time series trajectories of one component in "history" part.

3. For financial sector experiment, this set is combination of time series trajectories of selected financial sector components in "history" part.

3.3.1 The DAX Index

There are three experiments of DAX index components, the whole experiment, individual experiment and financial sector experiment. The results of experiments are visualized with the variance of log–return factors plot and confidence interval plots for chosen components. The confidence interval plots for chosen components would present the result of predictability for components with extreme variances.

3.3.1.1 Whole Experiment

The whole experiment is performed and let us set the parameter $k = 60$, "present" length of 60, "future" length of 60 and the distance measurement is Euclidean distance. The variances of "future" fragment predicted log–return factors against time plot for all 30 components are plotted within on graph. In this graph, it is easier to observe the trend of predicted log–return variances and the trend of the variances. Figure 3.5 shows variance of predicted log–return factors against time plot has linear trend from date 1 in "future" part time fragment to date 41. After this date, the variance plot has a horizontal trend and it converges to approximately 0.005 till the end of "future" part time fragment. For some components, the variance is relatively high (i.e. component "CBK"). It is observed that 1 cluster is formed from variances. The component "CBK" is considered as outlier as its variance is relatively far away from this cluster. This cluster contains variances of the rest of components and it seems that the variances have a general upward trend and the variances, in the end, lie within the range between 0.002 and 0.007. For the components with extreme value of variance , let us define the components with high variance are defined as "high variance" case. For those components with the variance that is closed to the average variance, let us define those components are the "average variance" case or neutral result for our prediction. (i.e. component "BAYN") For those components with relatively small variance, let us define them to be "small variance" case. (i.e. component "BEI") Let us choose component CBK to be "high variance" case, component BAYN to be "average variance" case, and component BEI to be "small variance" case. From Figure 3.6 it is observed that the "small variance" component has the best prediction results among these 3 components. The trend of predicted prices are extremely close to the real prices and all real prices lie in the confidence interval. For "average variance" component, some of the real prices lie outside the confidence interval and the trend of predicted prices are different if compared with

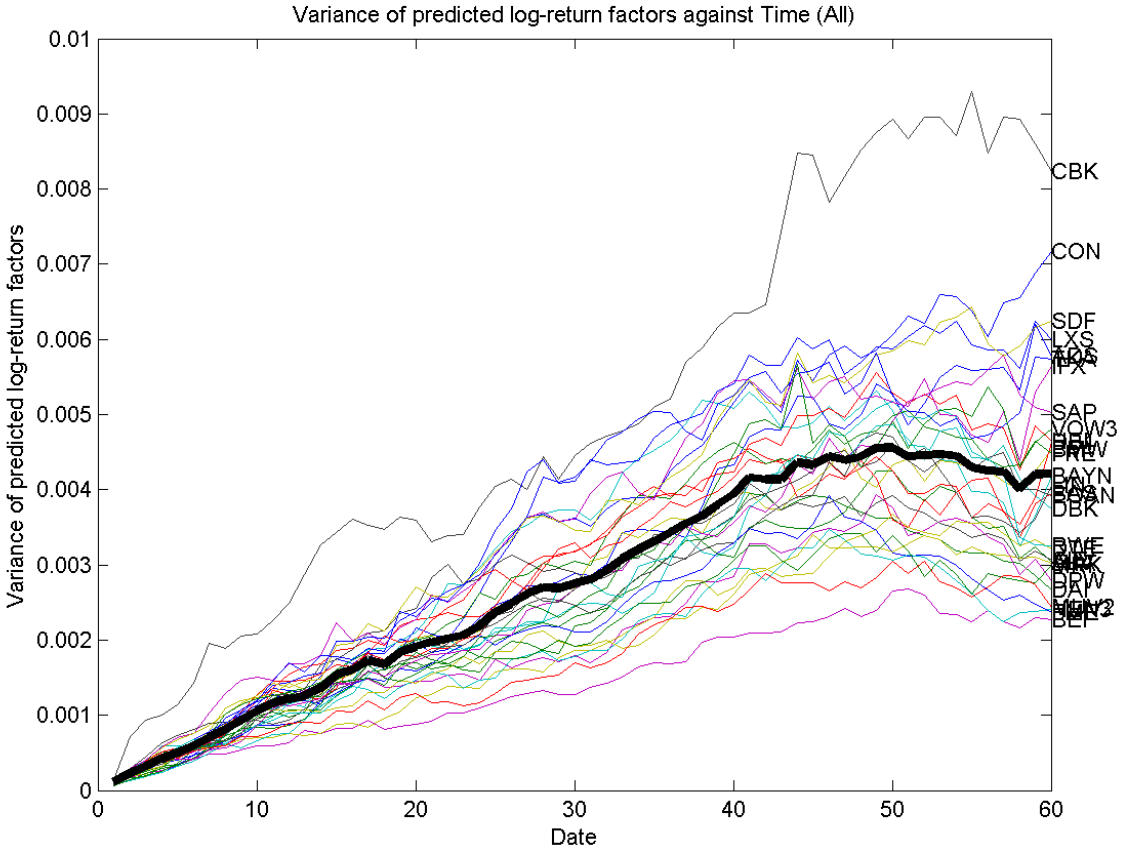
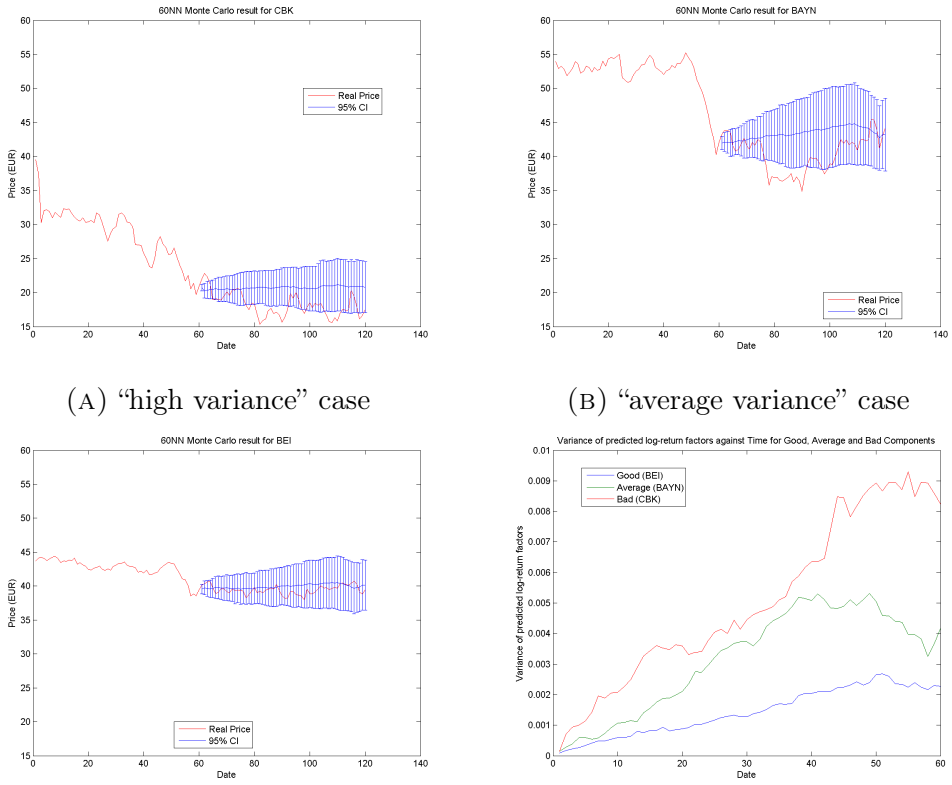


FIGURE 3.5: Graph of variance of predicted log–return factors against time for all 30 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 60 (Euclidean Distance). The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 30 components.

the real prices. The prices movement between date 75 and date 95 is not predicted since the real prices lie outside the confidence interval and the real prices have a general downward trend in this time interval. The "high variance" component has the worst result of distribution predictions. The average values of predicted prices have a general horizontal trend while the real prices have downward trend and almost 30% of real prices lie outside the confidence interval. By setting a longer length of "future" time fragment, the longer trend on prediction prices variance is analyzed. Let us set the length of "future" fragment to 100 when the other parameters remain the same. Using the same "history" and find the nearest neighbors of the same "present", the performance of the experiments on longer "future" is analyzed. Figure 3.7 presents the variances of log–return factors for "future" time fragment of all 30 components. The variance plots have upward linear trend from the beginning to date 41, then the variances have a horizontal



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) variance plot

FIGURE 3.6: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 60 for selected "high variance", "average variance" and "high variance" components (Euclidean Distance). Variance of predicted log–return factors against time for these components plot is shown in the end.

line for approximately 20 dates (this is similar to the variances plot for "future" part of length 60). There is an another upward linear trend till the end of the graph. The second upward linear trend has a larger gradient than the first one and it has an upward trend till the end of time fragment. It is also observed that in this variances plot, 2 clusters of variances are formed. The first cluster contains several components with relatively high variance while the second cluster contains the most of the components with relatively smaller variance.

Let us choose three components the same way as the experiment for setting "future" fragment to be 60. The "average variance" component has changed to component BAS and the "small variance" component has changed to component HEN3. The results are a bit worse than the experiment with "future" fragment of length 60. For "small variance" case, a small amount of the real prices lie outside the confidence interval and prediction result is bad from date 1 to date 110. This result becomes better from date 111 and the average predicted prices are very close to the

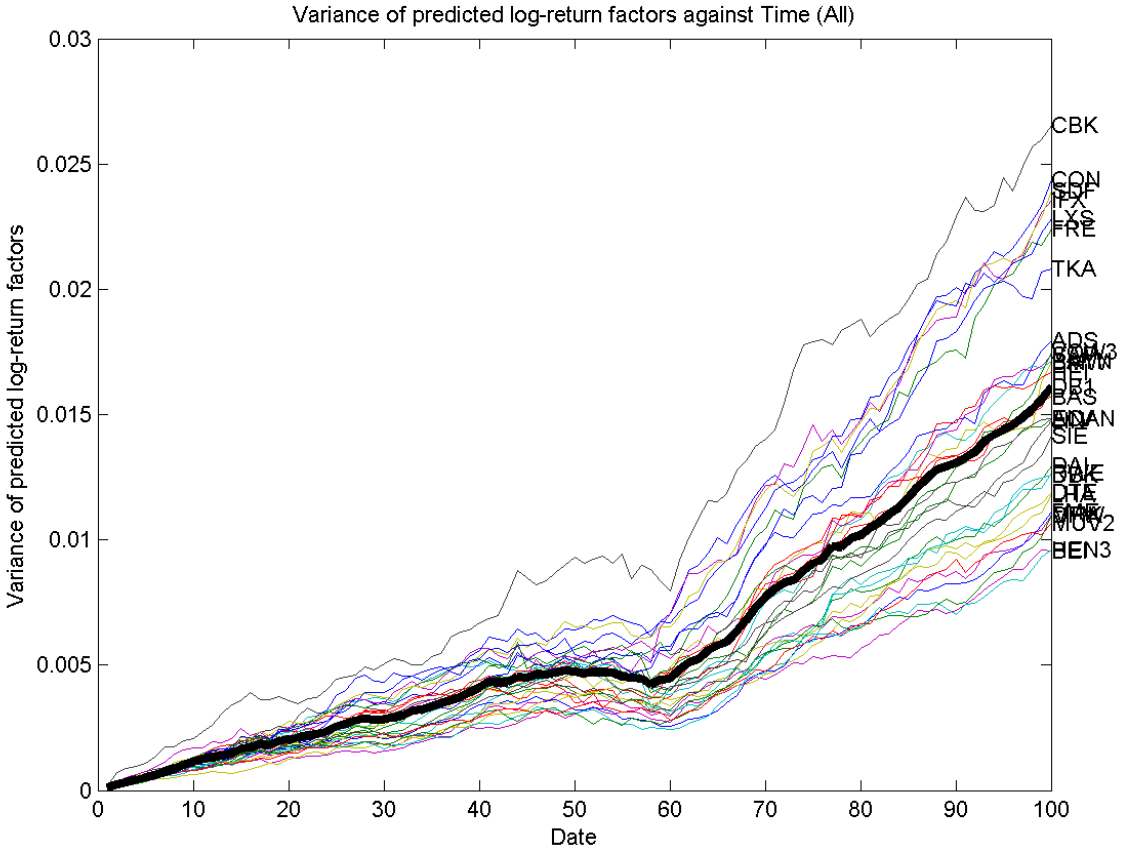
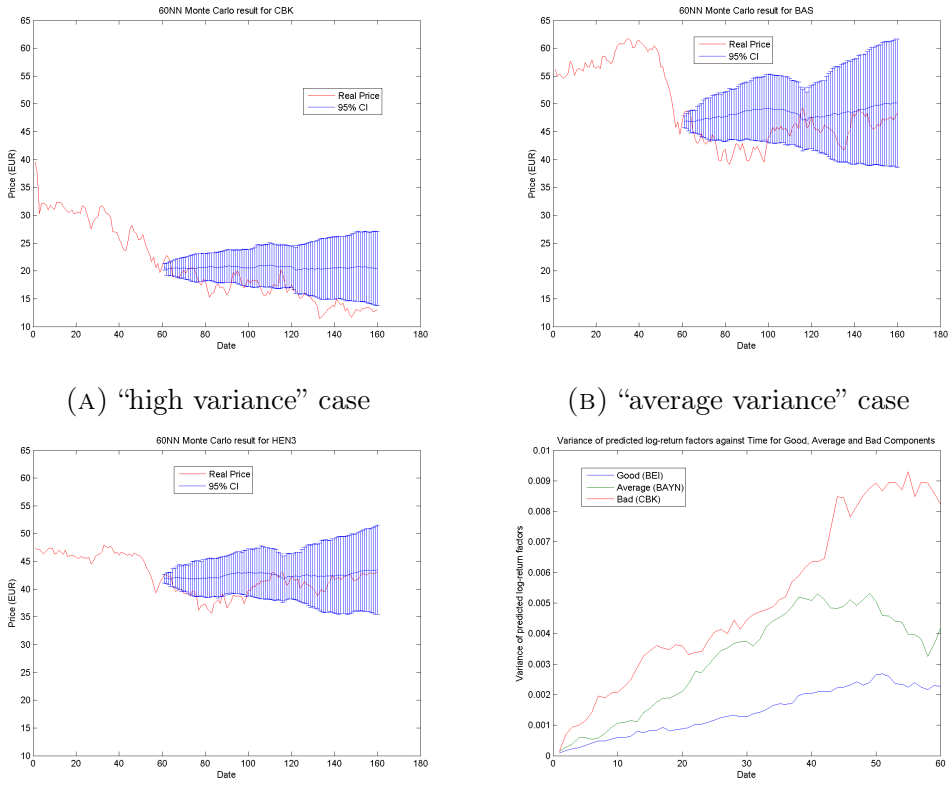


FIGURE 3.7: Graph of variance of predicted log–return factors against time for all 30 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 100 (Euclidean Distance). The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 30 components.

real prices. For "average variance" case, around 60% of the real prices lie outside of the confidence interval. Particularly, the real price from "future'âĂŸ fragment date 8 till "future" fragment date 40 lie outside of the confidence interval and the prediction result become better from date 41. For "high variance" case, most of the real prices lie outside of the confidence interval and the prediction result is worse than the first two cases. Compared with the results for "future" length 60, setting a smaller "future" time fragment would lead to a better prediction. In result of experiment with "future" length of 60 and 100, there is a horizontal trend from date 41 to date 60. Let us focus on the time interval of the first linear trend by setting the length of "future" to be 40. Figure 3.9 presents the variances of log–return factors of "future" part. It is observed that the variances form one cluster except for component CBK which has a relatively higher variance. All variances have a general linear trend from the beginning till the end. From Fig-



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) variance Plot

FIGURE 3.8: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 100 for selected "high variance", "average variance" and "high variance" components (Euclidean Distance). Variance of predicted log–return factors against time for these components plot is shown in the end.

ure 3.10, 3 components are selected for "high variance" case, "average variance" case and "small variance" case. Component MUV2 is selected to be "small variance" case, the predicted result is best among these 3 components. The most of the real prices lie within the confidence interval and the average predicted prices have similar trend in the beginning and the end. Component BMW is selected to be "average variance" case and component CBK is selected to be "high variance" case. For these two cases, most of real prices lie outside the confidence interval and the trend of real prices are quite different as the trend of average prices. The City Block distance is chosen for distance measurement for whole experiment. Let us set the parameters with $k = 60$, "present" fragment length 60, "future" fragment length 40. From figure 3.11, the variances have an upward linear trend in general most of the variances curves are quite closed to each other. It is observed that the variances of several components have slightly higher gradients. They forms approximately 2 clusters that the first cluster contains approximately 4 components with relatively higher variances and the second cluster contains the rest of

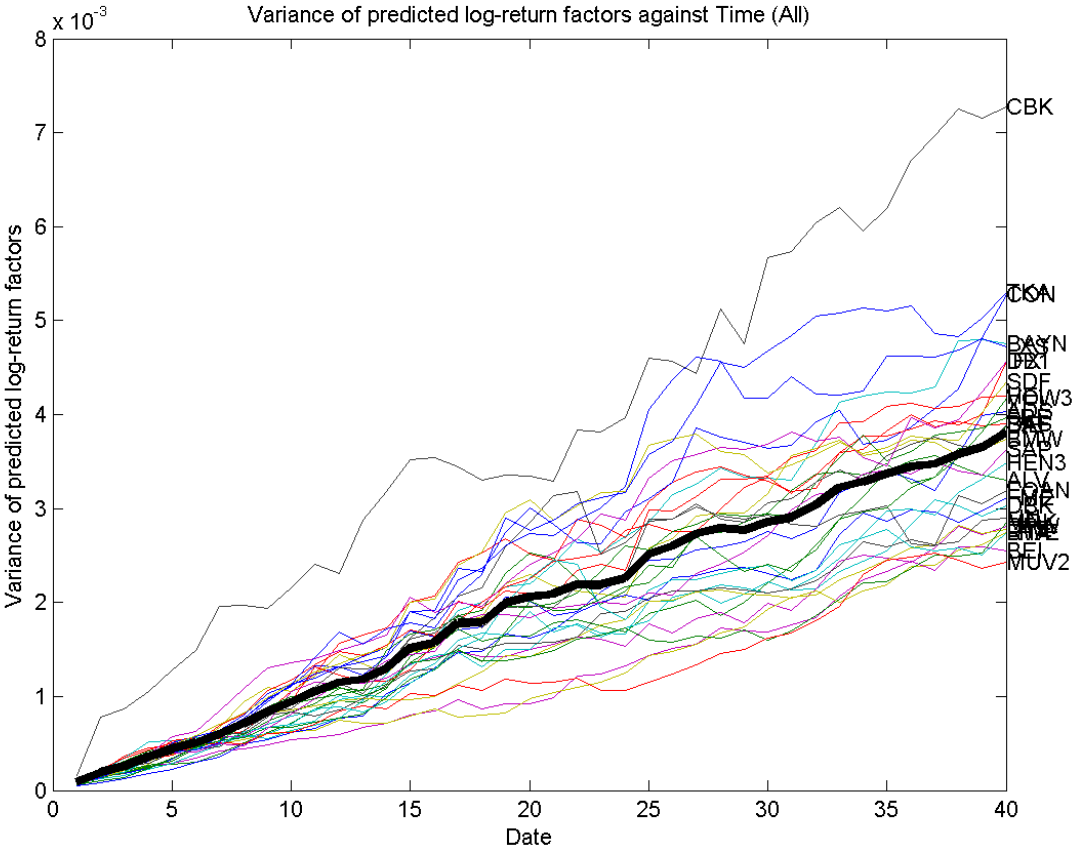
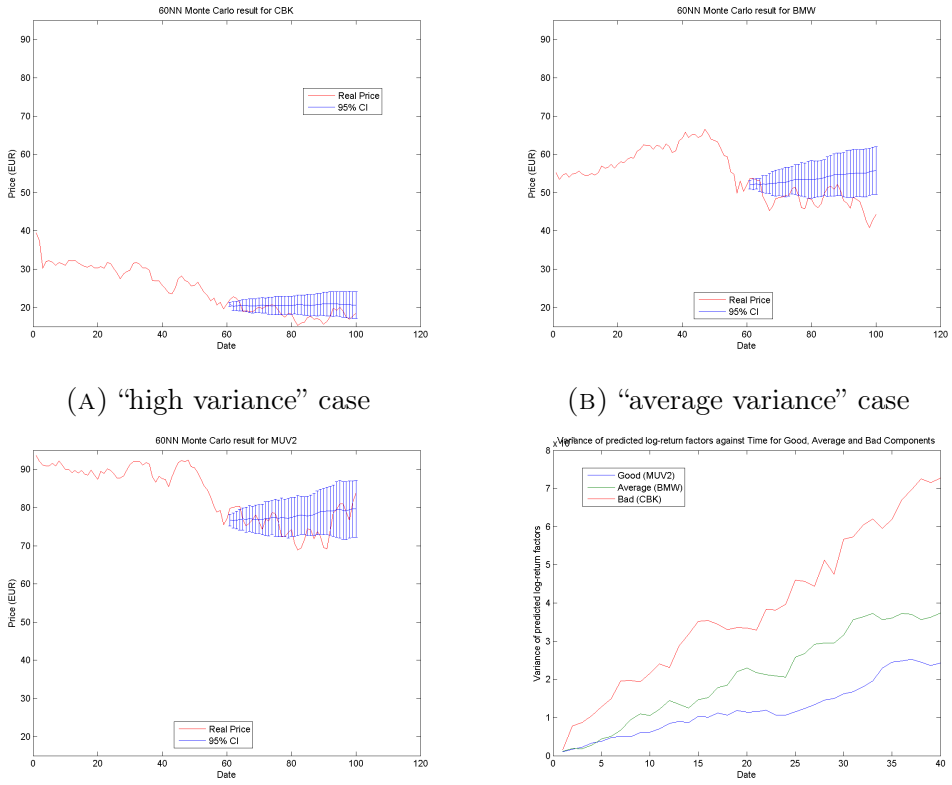


FIGURE 3.9: Graph of variance of predicted log–return factors against time for all 30 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (Euclidean Distance). The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 30 components.

components with variances closer to average variance. Let us select component CBK for "high variance" case, component RWE for "average variance" case and component DPW for "small variance" case. The "average variance" case has the best prediction results among these 3 cases. Approximately 25% of the real prices lie outside the confidence interval and the trend of the real prices in the end are predicted since the predicted prices are close to the real prices. For the "high variance" and "small variance" cases, more than 50% of real prices lie outside the confidence interval and the trend of real prices are not successfully predicted. The correlation distance is applied as the distance measurement for whole experiment. From Figure 3.13, the variances form one cluster only and all variances of components are close to the average variance. The general trend of variances for all components is linear upward trend. Let us choose component SIE for "high variance" case, component DPW for "average variance" case and component SDF for



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) variance Plot

FIGURE 3.10: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 40 for selected "high variance", "average variance" and "high variance" components (Euclidean Distance). Variance of predicted log–return factors against time for these components plot is shown in the end.

"small variance". The "small variance" case has the best result since most of real prices lie within the confidence interval and the trend of predicted prices and real prices are quite close. For "average variance" case, approximately half of real prices lie outside the confidence interval and the trend of real prices and predicted prices are not the same. For "high variance" case, all real prices lie within the confidence interval. However, the trend of the real prices and predicted prices are not close. The cosine similarity is applied as distance measurement for whole experiment. In Figure 3.15, the variances of all components form one cluster. The trend of the variances is a linear upward trend. The variances of all components are close to the average predicted prices. Let us choose component SIE for "high variance" case, component HEN3 for "average variance" case and component DB1 for "small variance" case. For these three cases, it is observed that the trend of real prices and the trend of predicted prices are not very close. However, for "high variance" case and "average variance" case, all real prices lie within the confidence intervals. The "small variance" case has the worst results among these three components

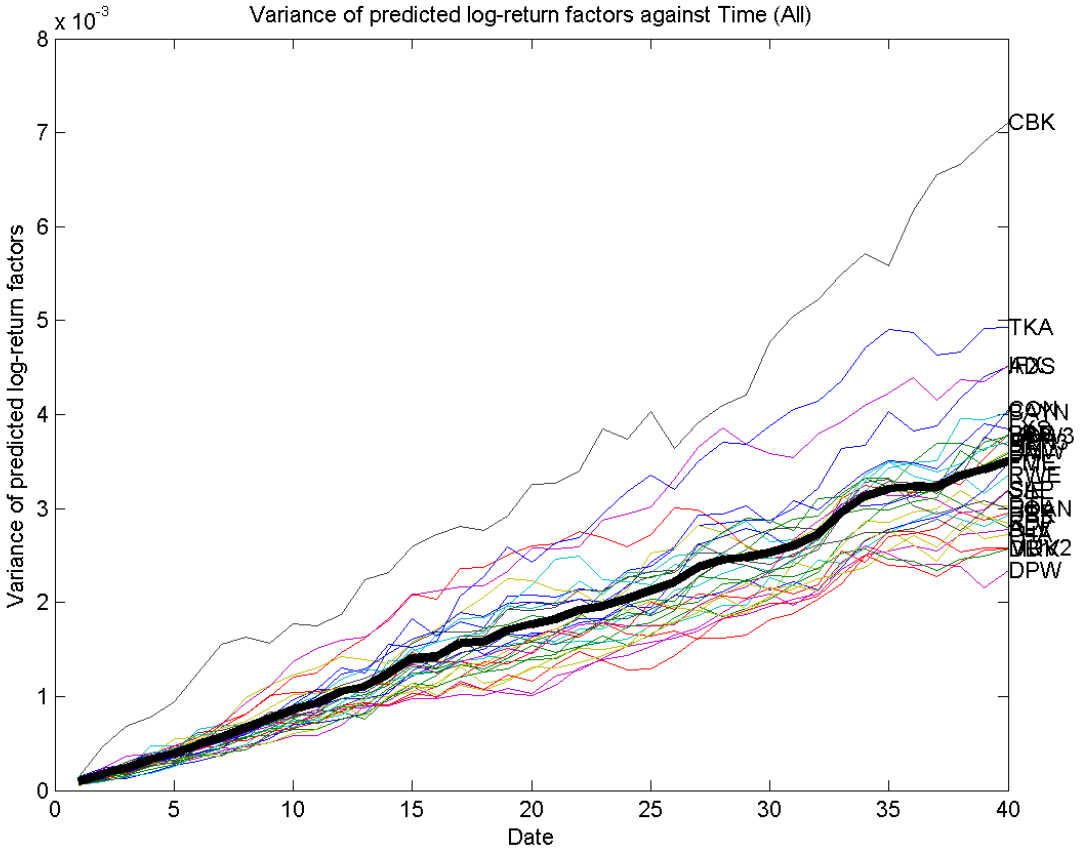
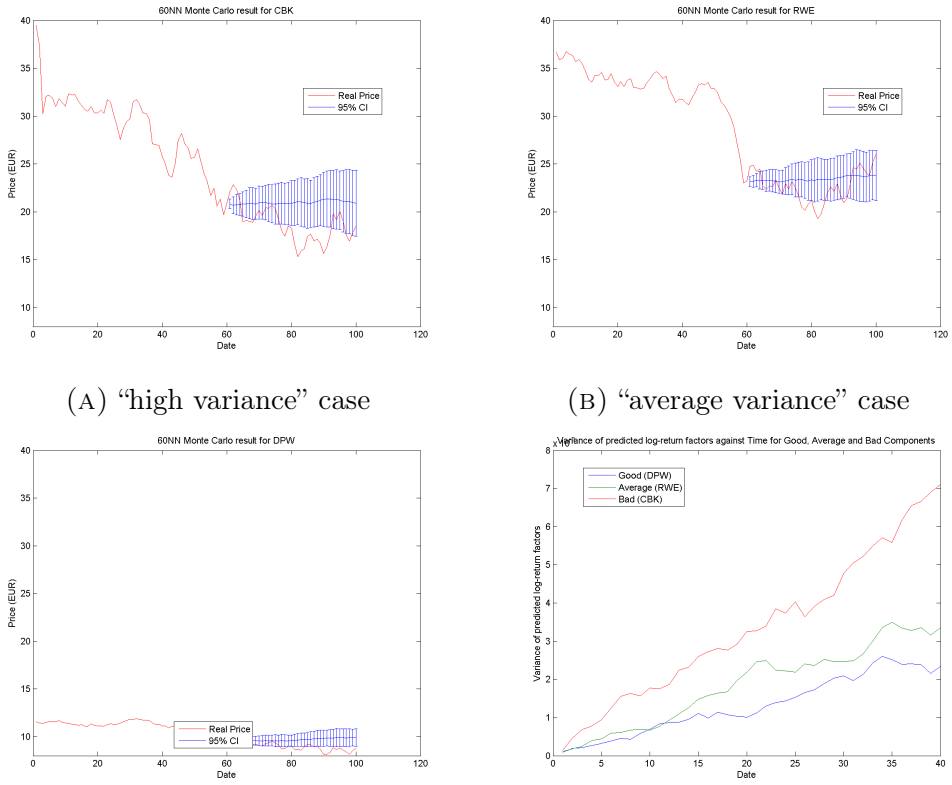


FIGURE 3.11: Graph of variance of predicted log–return factors against time for all 30 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (City Block Distance). The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 30 components.

but only quite a few of real prices lie outside the confidence interval. From the results, by analysis of three extreme components, the results of the experiment using Euclidean distance and correlation distance gives a better prediction on the prices. If comparing the value of variances for different distance measurements, it is observed that experiment using Euclidean and city block distance have relatively smaller variances than experiment using correlation distance and cosine similarity. It is also observed that the formation of the variances plots are similar between Euclidean distance case and City Block distance case while the formation of the variances plots are similar between the correlation distance case and cosine similarity case.



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) Variance Plot

FIGURE 3.12: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 100 for selected "high variance", "average variance" and "high variance" components (City Block Distance). Variance of predicted log–return factors against time for these components plot is shown in the end.

3.3.1.2 Individual Experiments

The individual experiment is performed since the change of "history" part is an important factor of our experiment. Let use set $k = 60$, "present" part of length 60, and "future" part of length 40 as parameters of individual experiments. The experiment using Euclidean distance is performed. Figure 3.17 represents the variances of predicted log–return factors for all components when choosing Euclidean distance as distance measurement. It is observed that the variances have a linear upward trend. The variances of several components are relatively high while the variances of most components are smaller. It is clear from the figures that 3 clusters of variances are formed and the average variance is located in the second cluster. In the first cluster, the components have large variances in general. In the second cluster, the components have variances which are close to the average variance. In the third cluster, the components have small variances in general. Let us choose component LXS for "high variance" case, component CBK for "average

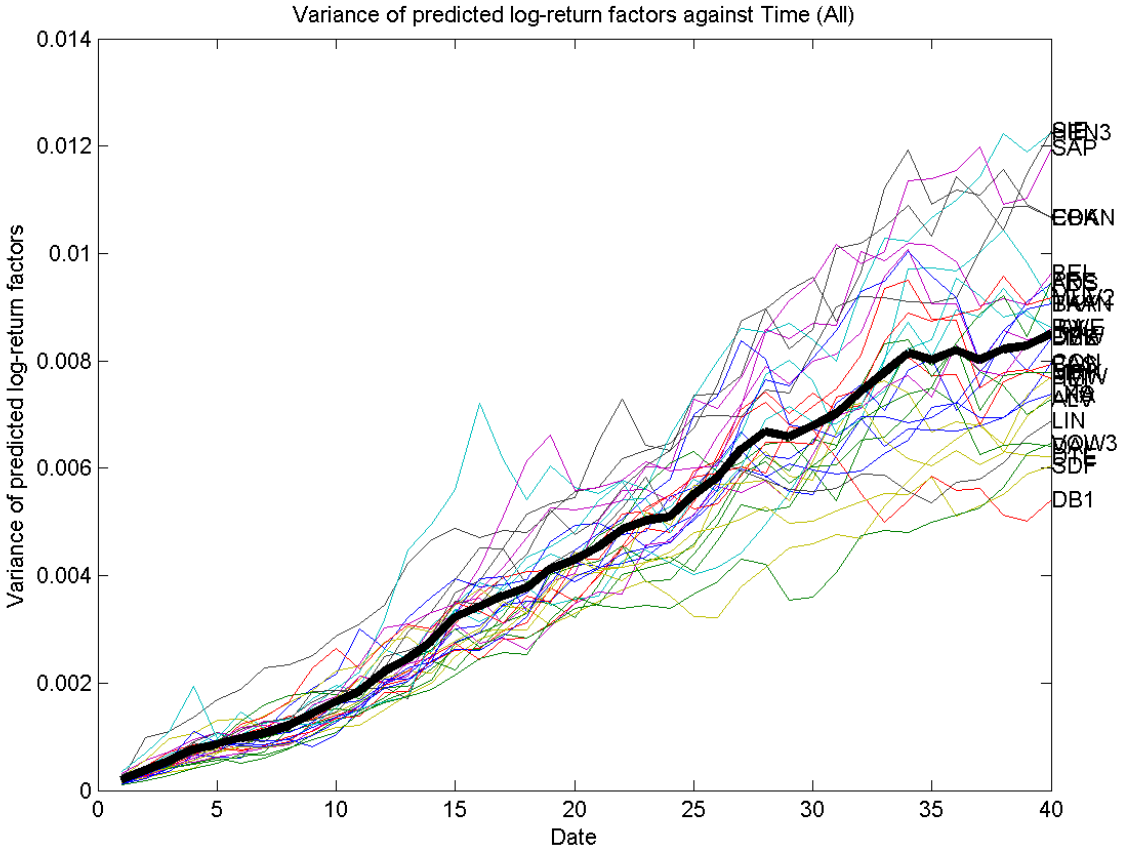
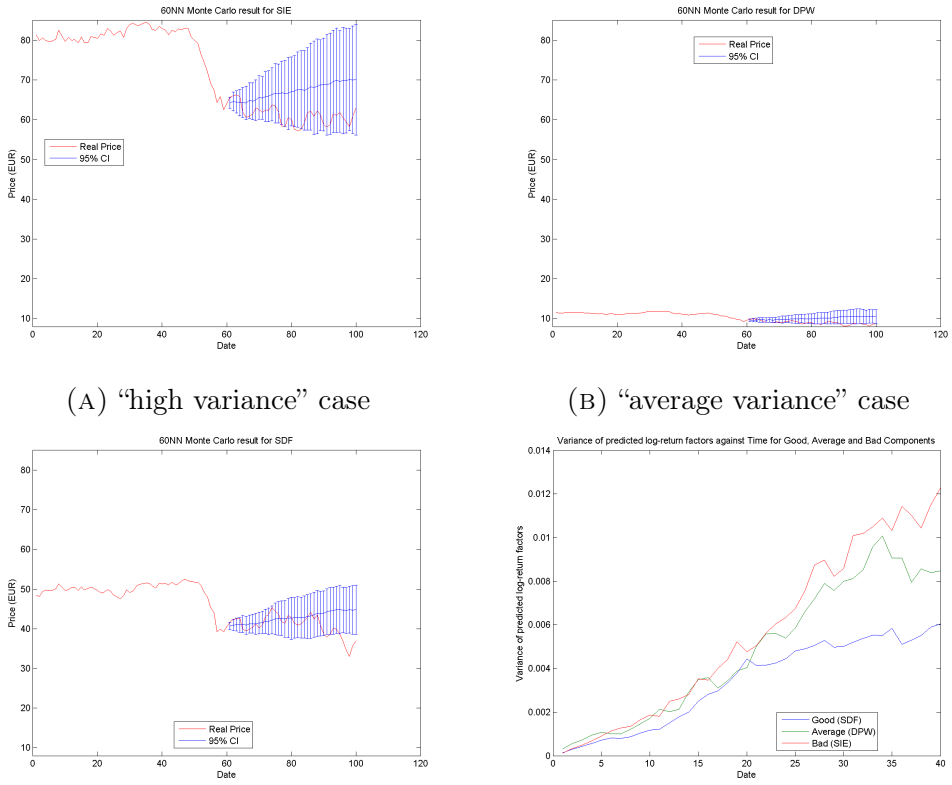


FIGURE 3.13: Graph of variance of predicted log–return factors against time for all 30 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (Correlation Distance). The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 30 components.

variance" and component MRK for "small variance" case. Figure 3.18 visualizes the confidence interval for selected components and the variances of these 3 components. For these 3 cases, it is observed that the trend of predicted prices is different to the trend of real prices. For "high variance case", approximately 25% of the real prices lie outside the confidence interval. The predicted prices have completely different trend as when the real prices fall, the predicted prices have a upward trend. For the "average variance" case, approximately 40% of real prices lie outside the confidence interval but the general trend of the predicted prices are similar to the real prices. For "small variance" case, the component has smallest variance among these 3 components. The result is similar to the "average variance" case with fewer real prices lie outside the confidence interval. The city block distance is applied as the distance measurement in this experiment. Unlike the results in the whole experiment, in the individual experiment with city block



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) Variance Plot

FIGURE 3.14: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 100 for selected "high variance", "average variance" and "high variance" components (Correlation Distance). Variance of predicted log–return factors against time for these components plot is shown in the end.

distance, the variances form 3 clusters. One cluster contains several components with high variances. The second cluster contains several components which is close to the average variance. The third cluster contains most of components and the variances are smaller than the average variance but they are closer to the average variance. Let us choose component LXS to be the "high variance" case, component SDF to be the "average variance" case, component MRK to be the "small variance" case. They are selected as representations for three clusters. For "high variance" case, the component has largest variances among these 3 components. There are about 30% of the real prices lying outside the confidence interval. It is observed that the predicted prices have a completely different trend. The real prices have a general upward trend while the real prices fall. For "average variance" case, the result is the best among these 3 cases. It has the smallest number of real prices lying outside the confidence interval. The trend of predicted prices are close to the real prices from the beginning, but in the end, the real prices drop when the

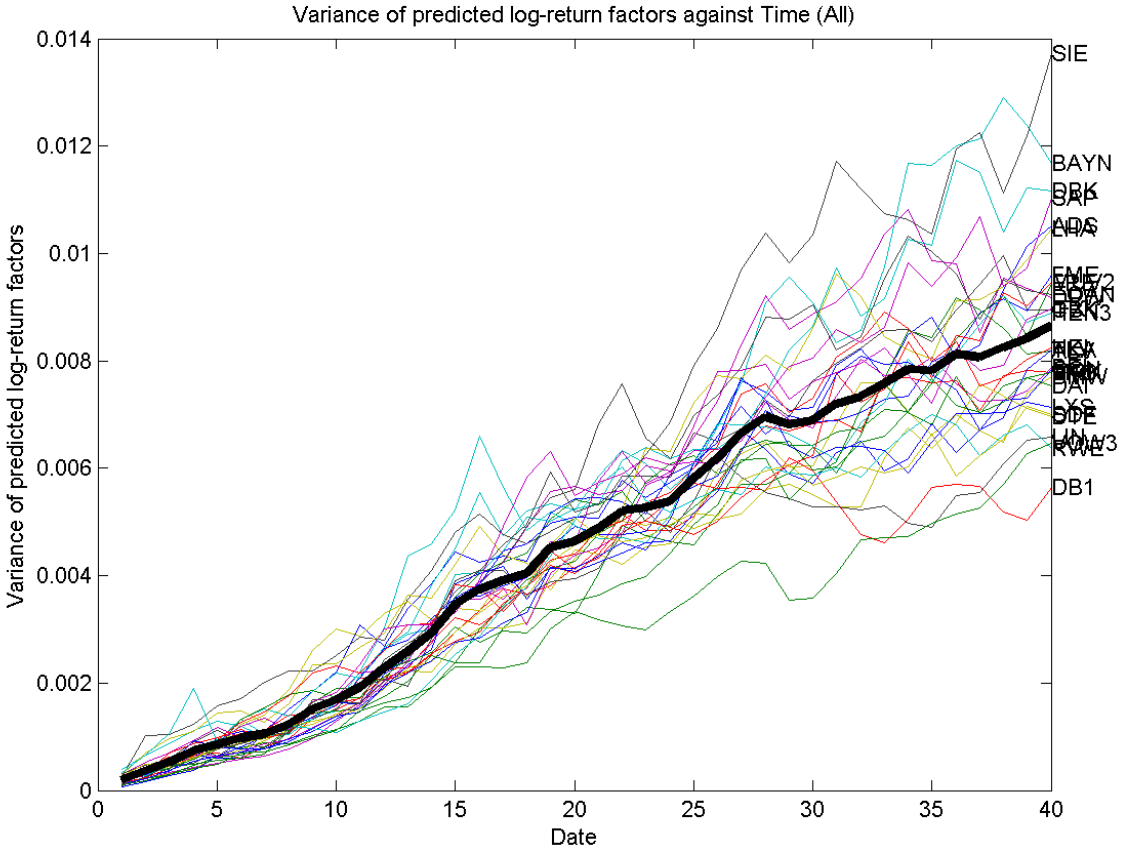
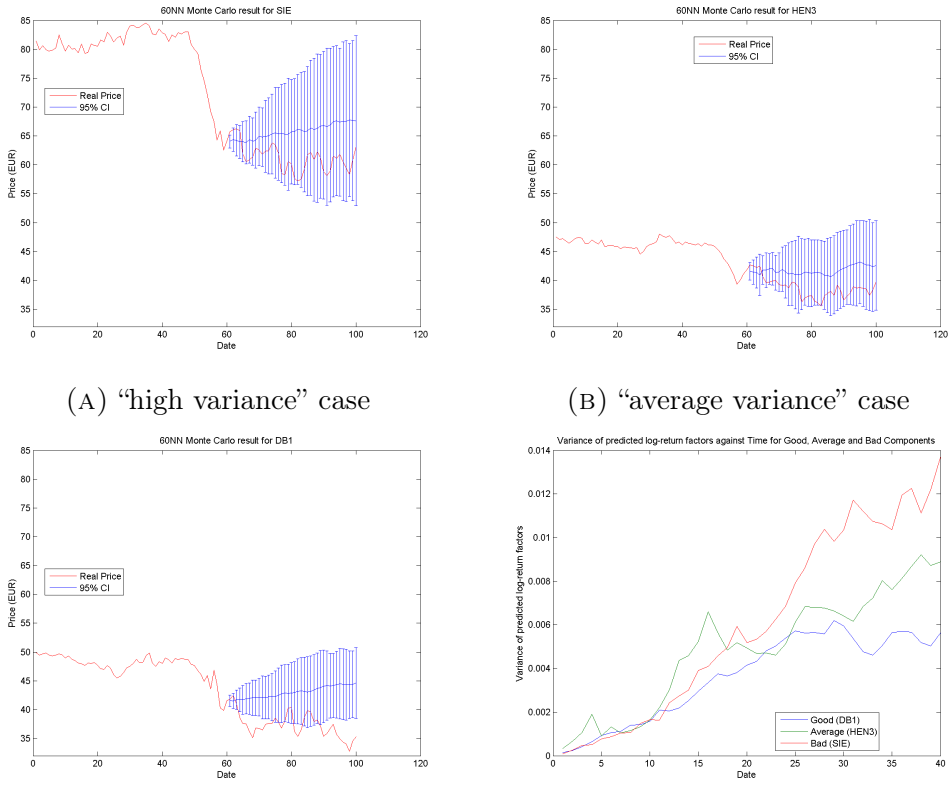


FIGURE 3.15: Graph of variance of predicted log–return factors against time for all 30 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (Cosine Distance). The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 30 components.

predicted prices still have an upward trend. For "small variance" case, approximately 30% of real prices lie outside of confidence interval and the trend of the predicted prices and the trend of real prices are different. The correlation distance is applied as the distance measure for an individual experiment. From Figure 3.21, 3 clusters of variance are formed in this experiment. The first cluster has several components with high variance. The second cluster has several components with variances that are close to the average variance. The third cluster contains most of the components with small variance. The second and third clusters are closer than the first cluster. Let us choose component VOW3 for "high variance" case, component BMW for "average variance" case and component DTE for "small variance" case. Figure 3.22 visualizes the confidence interval for selected components and the variances plot against time. For "high variance" case, most real prices lie within the confidence interval. However, the predicted prices are not the same as



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) Variance Plot

FIGURE 3.16: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 100 for selected "high variance", "average variance" and "high variance" components (Cosine Distance). Variance of predicted log–return factors against time for these components plot is shown in the end.

the real price since the real prices have downward trend while the predicted prices go upward. For "average variance" case, approximately half of the real prices lie outside the confidence interval. The predicted prices have different trend than the real prices. For "small variance" case, the variances are relatively much smaller than the other 2 cases. It is observed that most of the real prices lie within the confidence interval and the predicted prices have similar trend as the real prices in the beginning but they have an upward trend and the real prices have downward trend and an upward trend in the end. The cosine similarity is applied as the distance measurement for individual experiment. From Figure 3.23, the variances form only 2 clusters, one with high variances contains several components and the other cluster contains most components with smaller variances. The variances of all components have general linear upward trend. Let us choose component VOW3 to be the "high variance" case, component BMW to be the "average variance" case and component DTE to be the "small variance" case. Figure 3.24 visualizes the confidence interval for these selected components. For "high variance" case, only

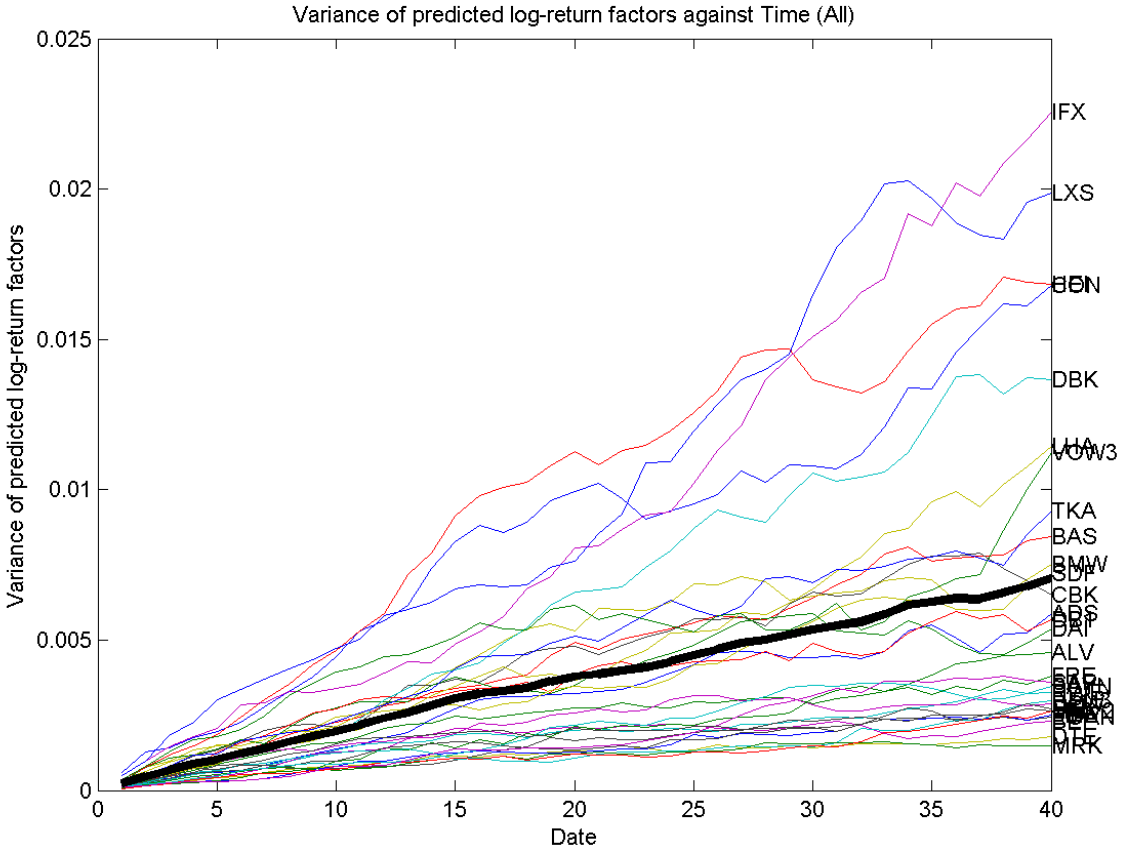
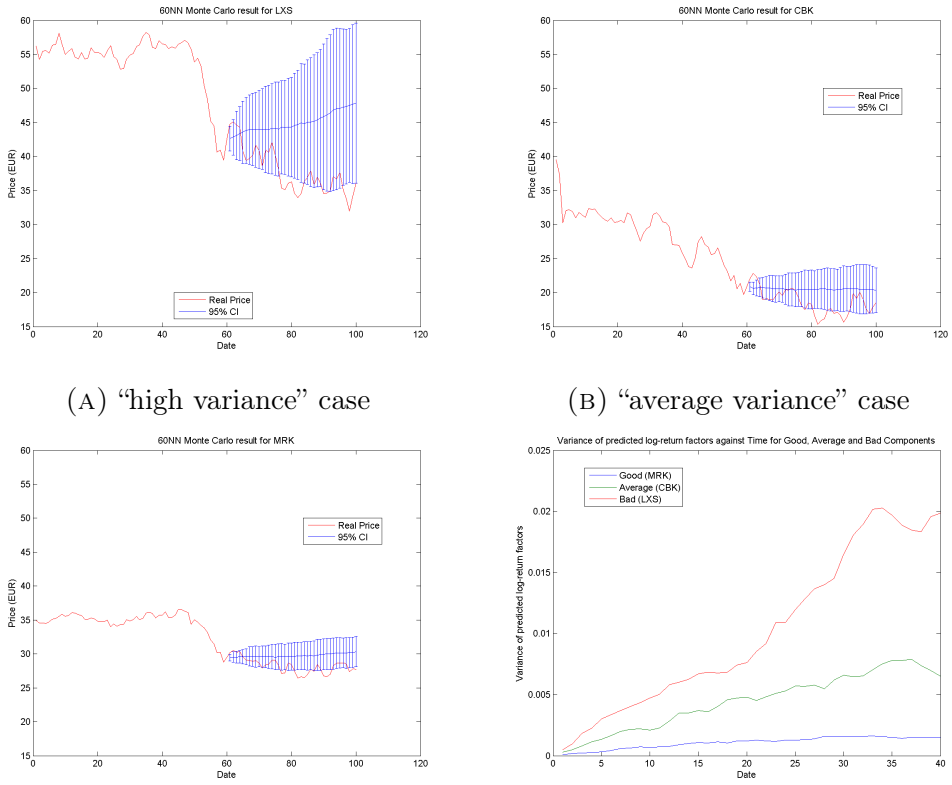


FIGURE 3.17: Graph of variance of predicted log–return factors against time for all 30 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (Euclidean Distance) for individual experiment. The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 30 components.

quite a few of the real prices in the end, lie outside the confidence interval. The trends of the real and predicted prices are different. For "average variance" case, approximately 20% of real prices lie outside the confidence interval and the trends of the real and predicted prices are different as well. For "small variance" case, most of the real prices lie within the confidence interval. The trends of the real and predicted prices are similar in the beginning, but this experiment fails to predict the trend of real prices from approximately date 75 (i.e. the real prices go down and then goes up while the predicted prices have a horizontal trend). From these results, the variances of the log–return factors plot of predicted prices are similar between experiment using Euclidean distance and city block distance and the variances plot are similar between experiment using correlation distance and cosine similarity. However, the prediction results of individual experiments with all distance measurement are not very good since the trends between predicted



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) Variance Plot

FIGURE 3.18: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 100 for selected "high variance", "average variance" and "high variance" components (Euclidean Distance) for individual experiment. Variance of predicted log–return factors against time for these components plot is shown in the end.

prices and real prices are quite different. The experiment results using Euclidean distance gives the best prediction among these four cases since the "average variance" case gives a good prediction while for the other distance measurements, the trends between predicted prices and real prices are different.

3.3.1.3 Experiment of Financial Sectors

The components of DAX index is categorized into several types (i.e.clothing, insurance, chemicals etc.). The components of the financial sector are mainly studied in this experiment. Table 3.1 lists the categories for all DAX components. Let us choose the components with category insurance, banking, and securities to be the components of the financial sector. The components chosen are ALV, CBK, DB1, DBK, and MUV2. In this experiment, the "history" part only contains log–return time fragments for these 5 components. The kNN algorithm is applied for

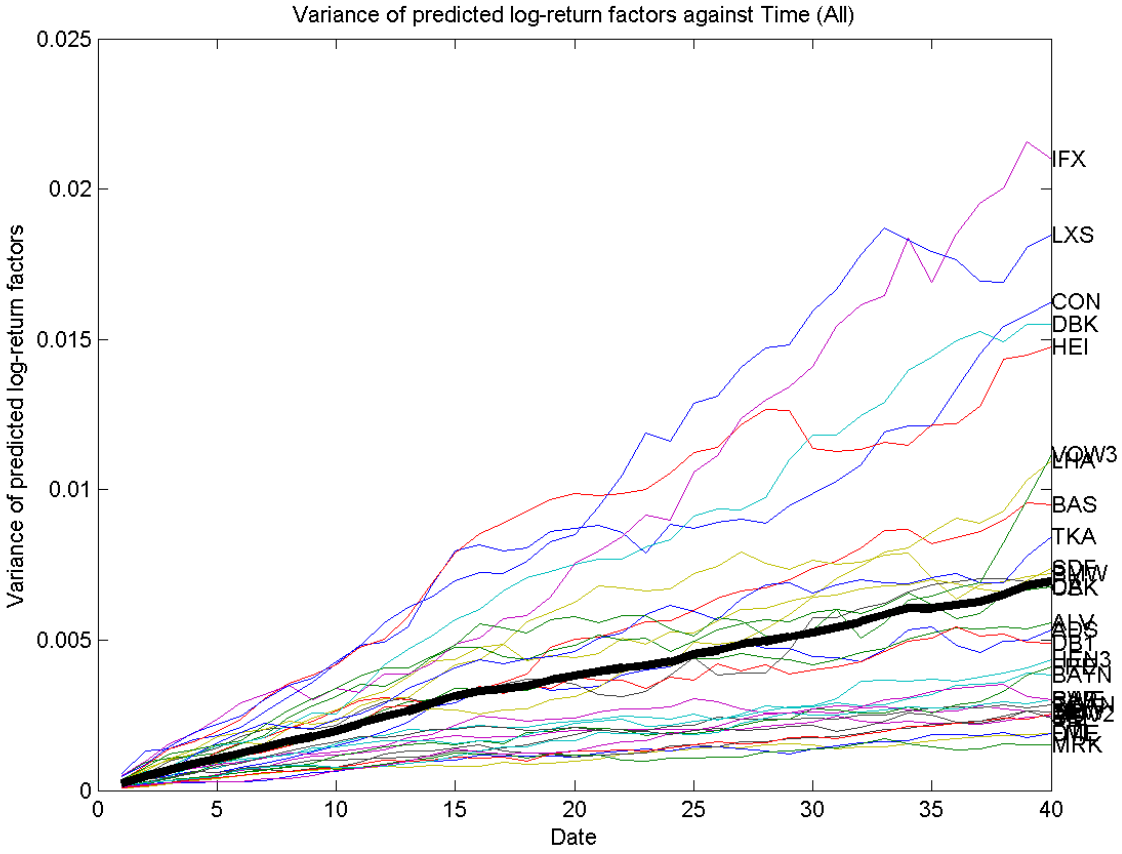
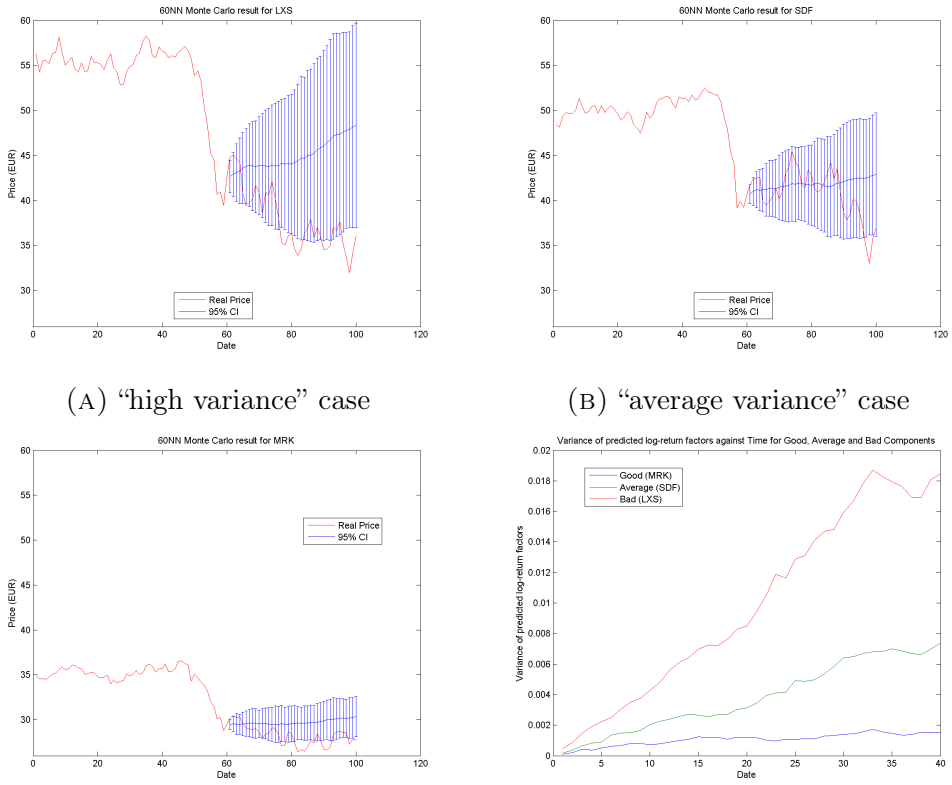


FIGURE 3.19: Graph of variance of predicted log–return factors against time for all 30 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (City Block Distance) for individual experiment. The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 30 components.

"present" part log–return time fragments of each one of 5 component, searching the nearest k neighbors from the new "history" time fragments. For the financial sector experiment, let us set $k = 60$, "present" part of length 60 and "future" part of length 40. Figure 3.25 is the combined variance graphs of selected 5 components for the individual experiment, whole experiment and financial sectors experiment. In general, the variances for individual experiment are higher than the variances for the other two experiments. Particularly, for component DBK, the variance for the individual experiment is about 7 times of the variances for other two experiment. It is also observed that the trend of variances for the whole experiment and financial sectors experiment are similar. Comparing the variances of the other 3 components (i.e. component ALV, DBK, and MUV2), the variances for the whole experiment are slightly smaller than the variances for financial sectors experiment. The variances for financial sectors experiment have a linear upward



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) Variance Plot

FIGURE 3.20: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 100 for selected "high variance", "average variance" and "high variance" components (City Block Distance) for individual experiment. Variance of predicted log–return factors against time for these components plot is shown in the end.

trend from date 1 till date 35. Then the variances have a horizontal trend from date 36 till date 40 (for several components, the variances have a small downward trend). Let us choose component DBK and component MUV2 for further analysis. From Figure 3.25, the variances of component DBK is extremely high for individual experiment while for component MUV2 the variances are relatively small. From Figure 3.26, component MUV2 has a better prediction result than component DBK. For component MUV2, the predicted prices have similar trend with the real prices in the beginning and the end, and approximately 20% of real prices lie outside the confidence interval. For component DBK, only quite few real prices lie within the confidence interval and the predicted prices have completely different trend as the real prices. The city block distance is used as the distance measurement for financial sector experiment. From Figure 3.27, the variances have a linear trend. Comparing with the variance plot of individual experiment, the formation of variances plot is similar. The variances for individual experiment

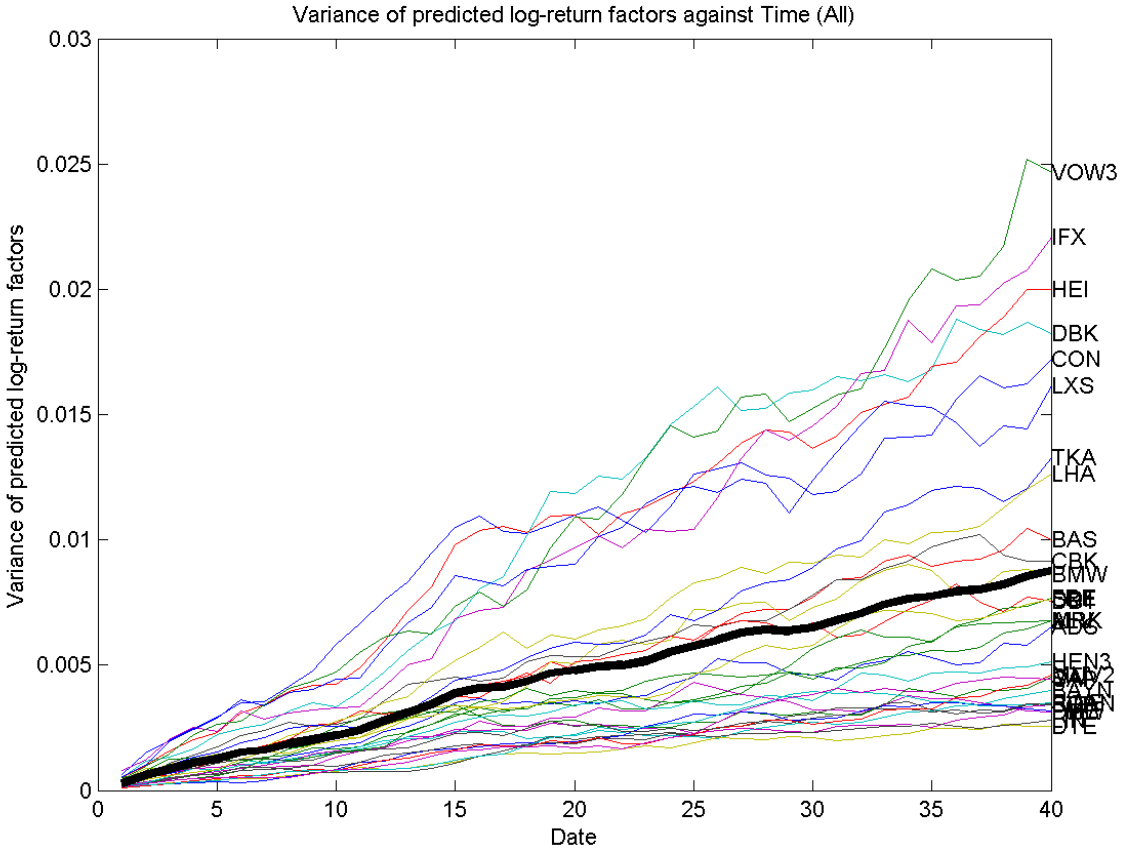
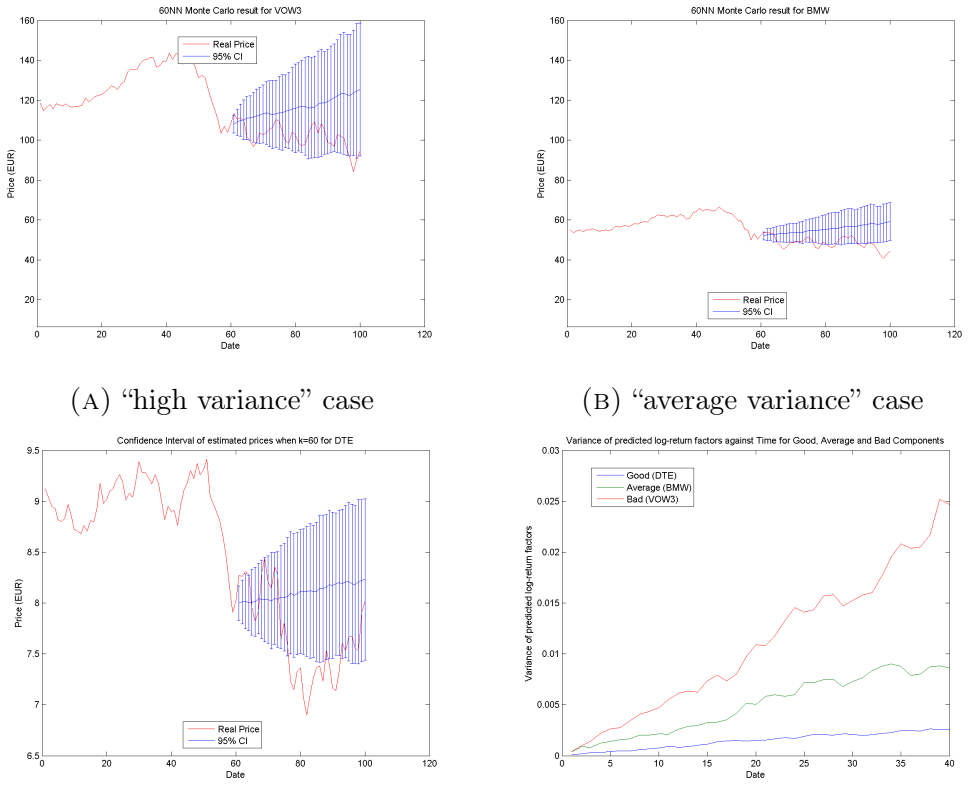


FIGURE 3.21: Graph of variance of predicted log–return factors against time for all 30 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (Correlation Distance) for individual experiment. The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 30 components.

are higher than the variances for independent experiment and whole experiment. Component DBK is selected since its different experiment results in individual experiment and component MUV2 is selected since the variances are the lowest for all experiments. From Figure 3.28, for component DBK, the predicted prices have different trend as the real prices and most of the real prices lie outside the confidence interval. For component MUV2, the prediction is good in the beginning and in the end and most of real prices lie within the confidence interval. For both component, this method failed to predict the double bottom pattern. For case of using correlation distance, it is observed from Figure 3.14 that the variances have a general linear trend and the variances for all 5 components are close to the average variances comparing to the whole experiment and individual experiment. For component DBK, the variance plot in financial sector experiment is different comparing to the variance plot in individual experiment as the variances



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) Variance Plot

FIGURE 3.22: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 100 for selected "high variance", "average variance" and "high variance" components (Correlation Distance) for individual experiment. Variance of predicted log–return factors against time for these components plot is shown in the end.

are smaller. The variances of other components in financial sector experiment are a bit higher than the variances in other two experiments. Component DBK and MUV2 is chosen as two examples in visualization of result in confidence interval plot. From Figure 3.30, component DBK has worse result than component MUV2. For component DBK the real prices in the beginning and in the end are included in the confidence interval in financial sector experiment. For component MUV2, all real prices are included in the confidence interval and this gives a perfect result. For the experiment using cosine similarity, from Figure 3.31 the variances in financial sector experiment have general linear upward trend with a bit higher gradient than the whole and individual experiments. The variances of all components are close to the average variance and component DBK is observed have difference in the variances if comparing with the plot in financial sector experiment and the individual experiment. For the rest of components, the variances in financial sector experiment are similar to the variances in whole experiment. Components DBK

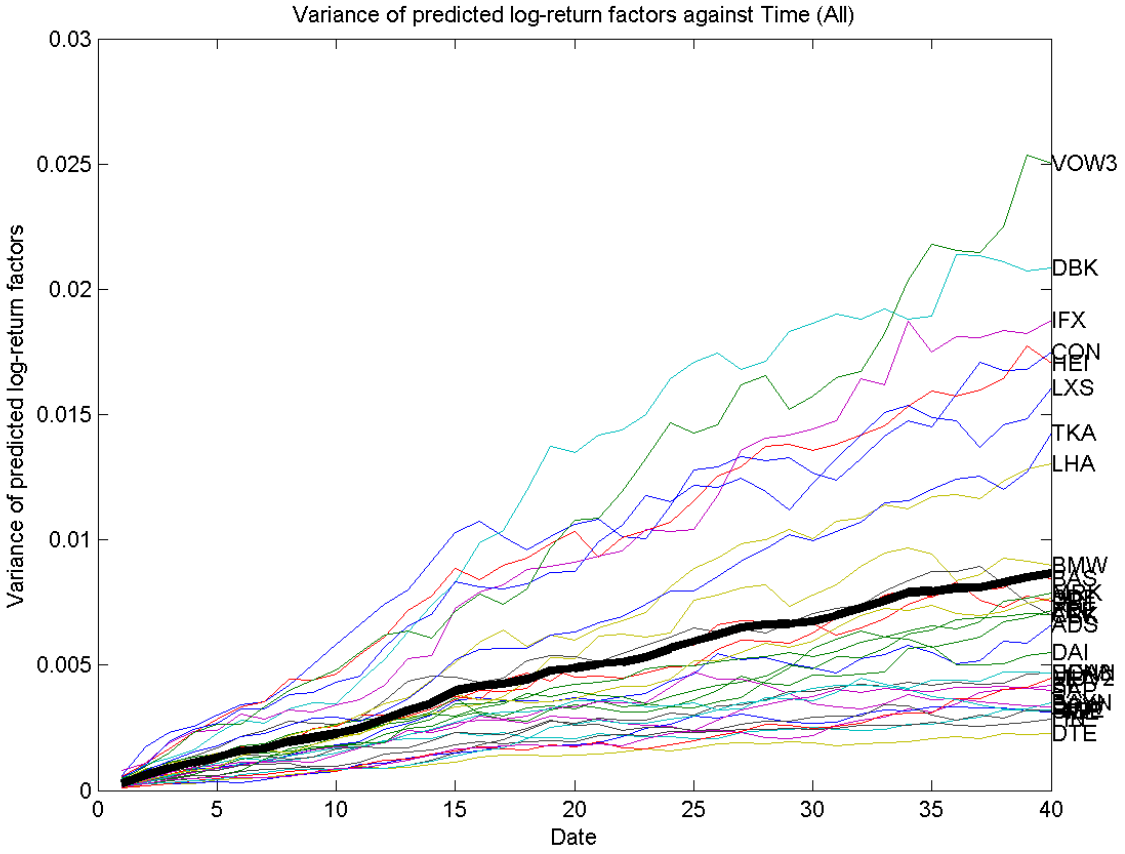
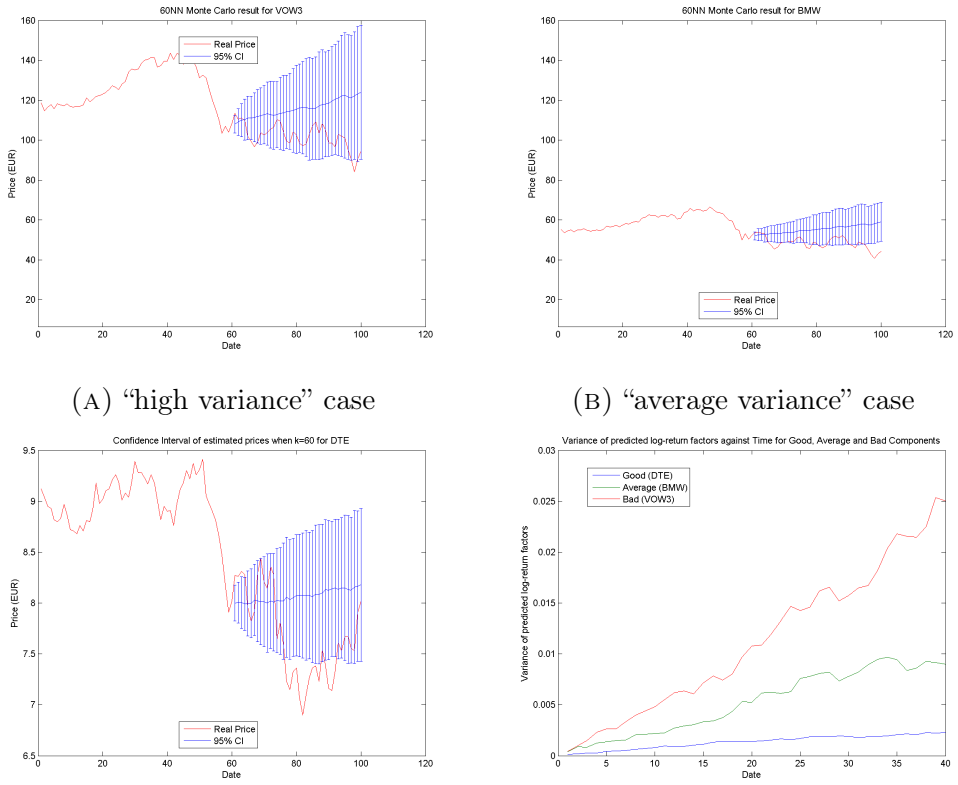


FIGURE 3.23: Graph of variance of predicted log–return factors against time for all 30 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (Cosine Distance) for individual experiment. The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 30 components.

and MUV2 are selected as 2 cases to visualize the result of financial sector experiment using confidence interval plot. From Figure 3.32, it is clear that the result for the case of component DBK is slightly worse than the result for the case of component MUV2. For component DBK, approximately 30% of real prices from date 70 to date 95 lies outside the confidence interval. While for component MUV2, all real prices lie within the confidence interval hence this result is perfect. Hence, it is concluded that for selected 5 components, the variances of most components (except for component DBK) are similar to the variances of individual experiment and whole experiment. From the visualization of results using confidence interval plots, the selected 2 components have different performance. For all 4 types of distance measurements, the result in individual experiment is the best among the three experiment for component DBK. However, for component MUV2, the result in financial sector experiment is the best. In general, it could be concluded that



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

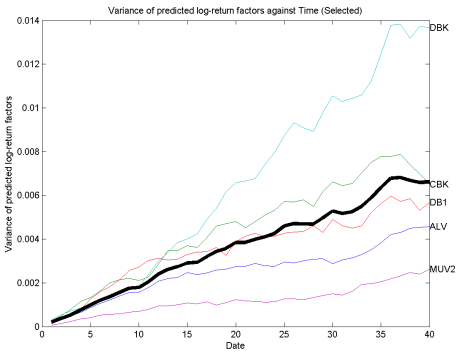
(D) Variance Plot

FIGURE 3.24: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 40 for selected "high variance", "average variance" and "high variance" components (Cosine Similarity) for individual experiment. Variance of predicted log–return factors against time for these components plot is shown in the end.

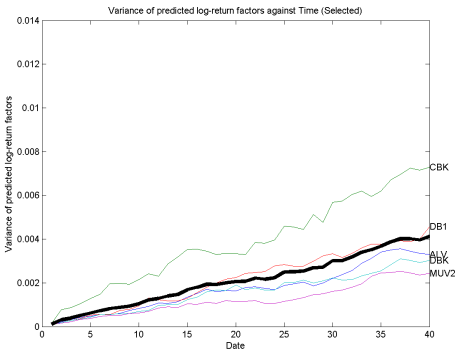
changing "history" part could lead to a better performance in this data.

3.3.2 The FTSE100 index

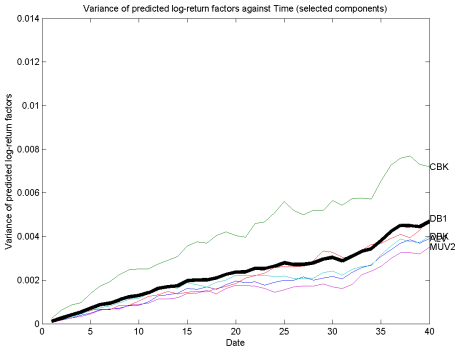
After running the program on components of DAX index, it is interesting to see the results of experiments for FTSE100 index. The financial sector experiment is performed for selected components of FTSE100 index. In this experiment, 18 components in financial sector are chosen. Similar to experiments using data from DAX index, the variances of predicted "future" part log–return factors and confidence interval of closing prices plots are used to visualize the results of experiment.



(A) Individual Experiment



(B) Whole Experiment

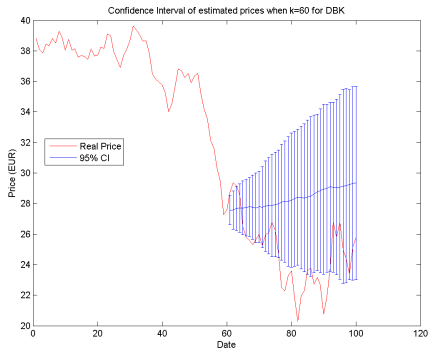


(C) Financial Factors Experiment

FIGURE 3.25: Graph of variance of predicted log–return factors against time for selected 5 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (Euclidean Distance) for individual experiment, whole experiment and financial factors experiment. The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 5 components.

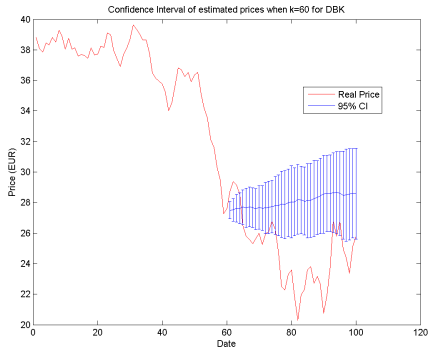
3.3.2.1 Financial Sector Experiment

In the financial sector experiment, 19 components with categories such as insurances (life or non–life), bank and financial services are chosen initially. Due to huge amount of missing values on closing prices, 1 component is ignored and the closing prices of 18 components listed in table 3.2 are used as raw data in this experiment. Similar to experiments of DAX index, four different distance measurements are applied to test this experiment. The variances plot of log–return factors for all components and the confidence interval plots for 3 selected components are used to visualize the result. In Figure 3.33, Euclidean distance is applied in the experiment. The variances form a general linear upward trend and the variances form one cluster except for component RSA. Hence for most of components, the variances are close to the average variance and the changes of the log–return factors of closing prices are similar. Let us choose component STJ for "high vari-

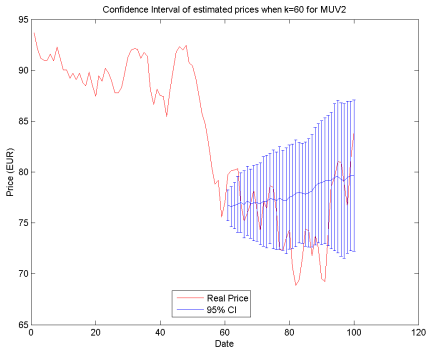


(A) Individual Experiment

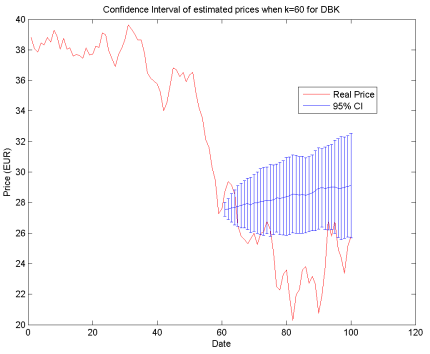
(B) Individual Experiment



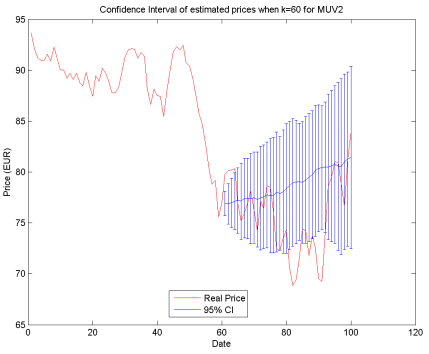
(C) Whole Experiment



(D) Whole Experiment

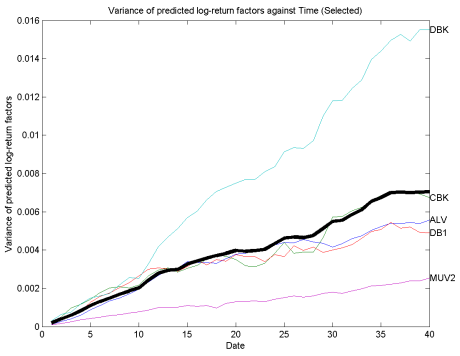


(E) Financial Sectors Experiment

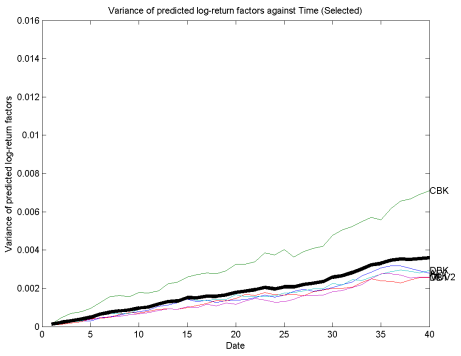


(F) Financial Sectors Experiment

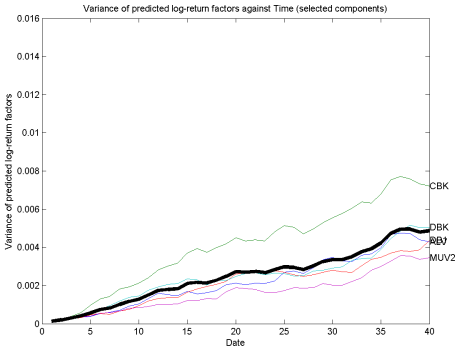
FIGURE 3.26: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 40 for component DBK and MUV2 (Euclidean Distance) for individual experiment, whole experiment and financial sectors experiment. Variance of predicted log–return factors against time for these components plot is shown in the end.



(A) Individual Experiment



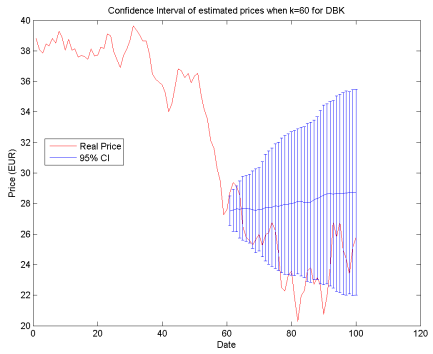
(B) Whole Experiment



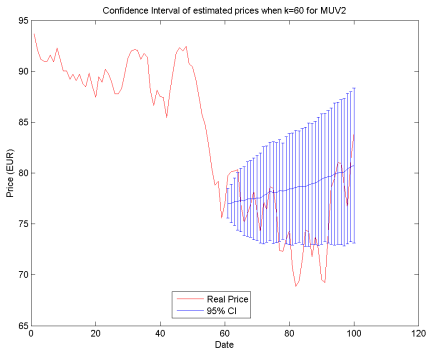
(C) Financial Factors Experiment

FIGURE 3.27: Graph of variance of predicted log–return factors against time for selected 5 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (City Block Distance) for individual experiment, whole experiment and financial factors experiment. The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 5 components.

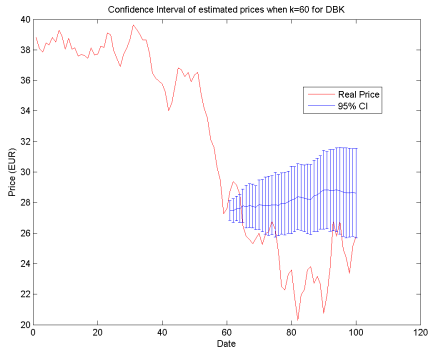
ance" case, component HSBA for "average variance" case and component RSA for "small variance" case. From Figure 3.34, it is observed that the result of "high variance" case is best among these 3 components since all real prices lie within the confidence interval . For "average variance" and "small variance" case, the results are similar and approximately 80% of real prices lie within the confidence interval since the experiment fails to predict the downward trend of prices in the end. It is interesting that the trend of real closing prices of "future" part for these two components are quite similar hence this experiment is giving a similar prediction result. City Block distance is then applied in this experiment. From Figure 3.35, the variances form a main cluster with one component has high variances and two components have small variances. The variances form a general linear upward trend and the average variance is positioned in the centre of main cluster. Let us choose component RBS for "high variance" case, component HSBA for "average variance" case and component BARC for "small variance" case. From the figure



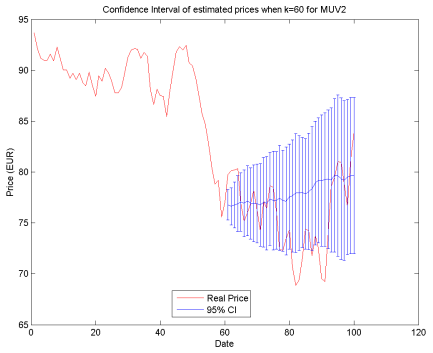
(A) Individual Experiment



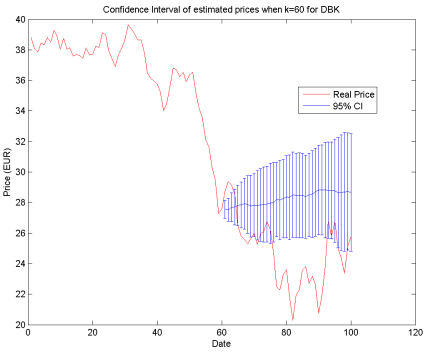
(B) Individual Experiment



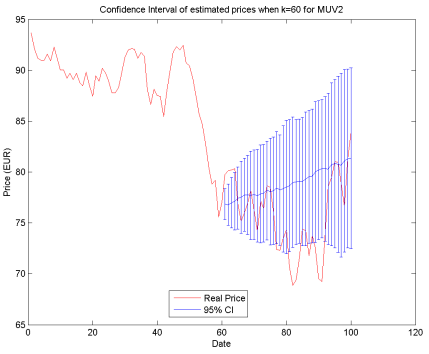
(C) Whole Experiment



(D) Whole Experiment

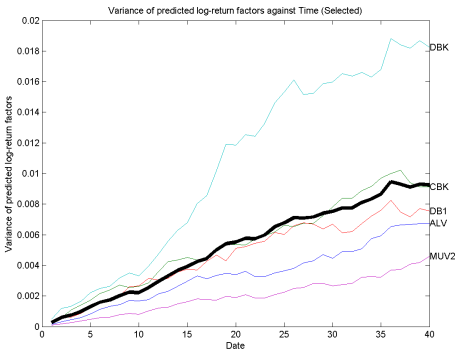


(E) Financial Sectors Experiment

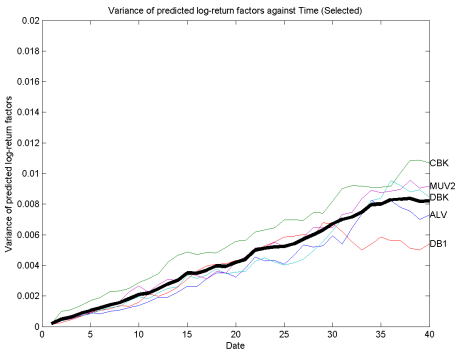


(F) Financial Sectors Experiment

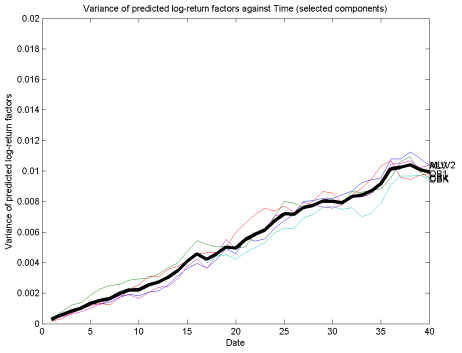
FIGURE 3.28: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 40 for component DBK and MUV2 (City Block Distance) for individual experiment, whole experiment and financial sectors experiment. Variance of predicted log–return factors against time for these components plot is shown in the end.



(A) Individual Experiment



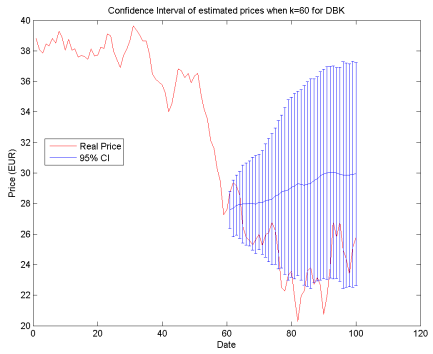
(B) Whole Experiment



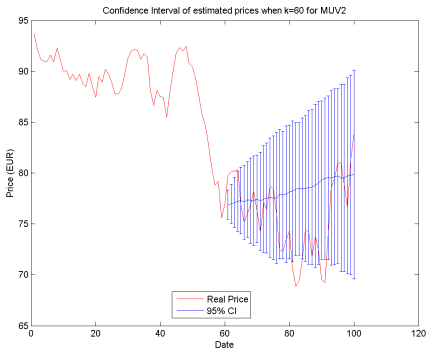
(C) Financial Factors Experiment

FIGURE 3.29: Graph of variance of predicted log–return factors against time for selected 5 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (Correlation Distance) for individual experiment, whole experiment and financial factors experiment. The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 5 components.

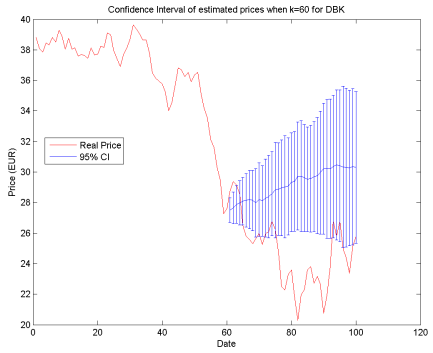
of confidence interval 3.36, it is observed that the results for "high variance" and "small variance" case are good since all real prices lie within the confidence interval. For "average variance" case, this experiment failed to predict the downward trend in the end of "future" part and approximately 15% of real prices lie outside the confidence interval. The correlation distance is applied in the experiment. Figure 3.37 represents the variances of log–return factors and from this figure, approximately 2 clusters are formed. One cluster contains higher variances have 6 components. Comparing to the average variance, they have higher variances in the beginning and there is a small fall at around date 20. The other cluster contains the rest of components with low variances in the beginning and unusual peak at around date 13. In general, the variances have a linear upward trend and the average variance is in between of two clusters and it has a general linear upward trend but with a small spike at around date 13. Let us choose component RBS for "high variance" case, component LGEN for "average variance" case and component III for small



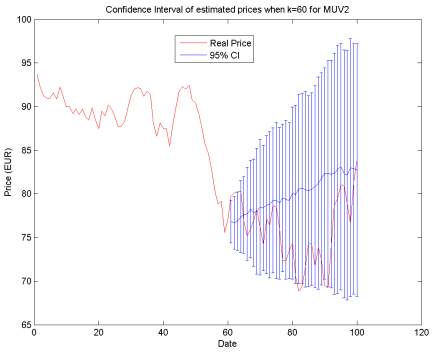
(A) Individual Experiment



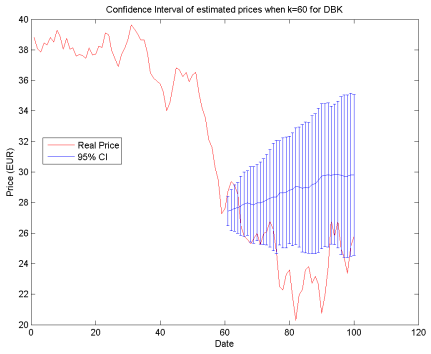
(B) Individual Experiment



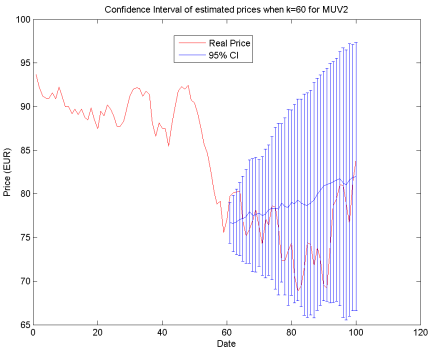
(C) Whole Experiment



(D) Whole Experiment

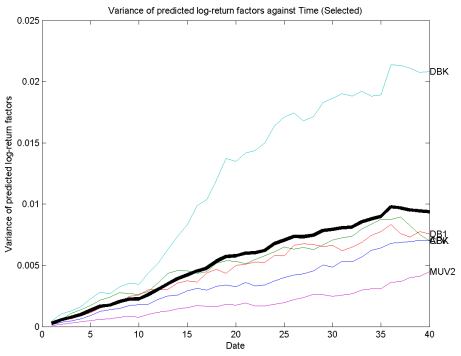


(E) Financial Sectors Experiment

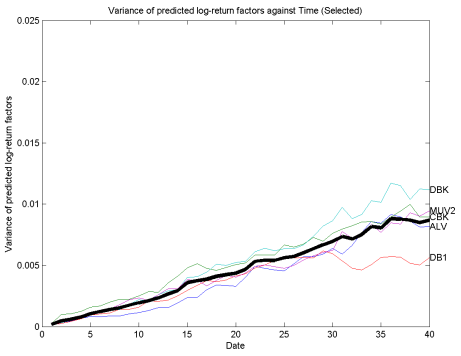


(F) Financial Sectors Experiment

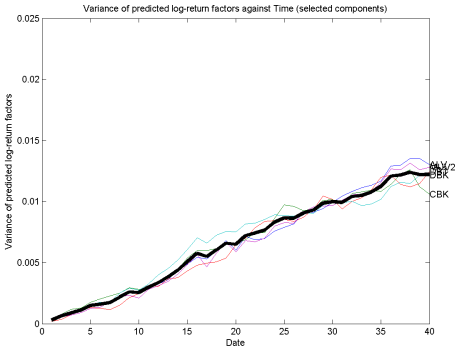
FIGURE 3.30: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 40 for component DBK and MUV2 (Correlation Distance) for individual experiment, whole experiment and financial sectors experiment. Variance of predicted log–return factors against time for these components plot is shown in the end.



(A) Individual Experiment



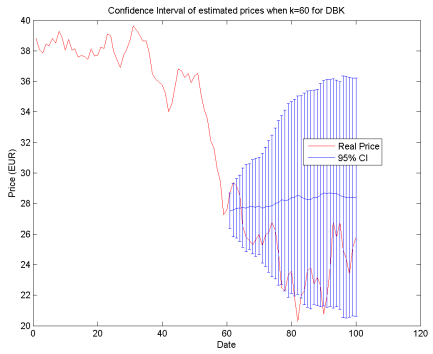
(B) Whole Experiment



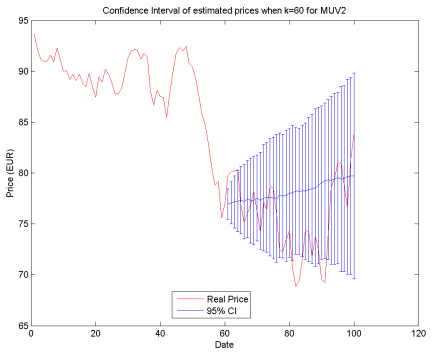
(C) Financial Factors Experiment

FIGURE 3.31: Graph of variance of predicted log–return factors against time for selected 5 components when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (Cosine Similarity) for individual experiment, whole experiment, and financial factors experiment. The name of component is presented on the last variance and the thick black line represents the average variance of these factors of 5 components.

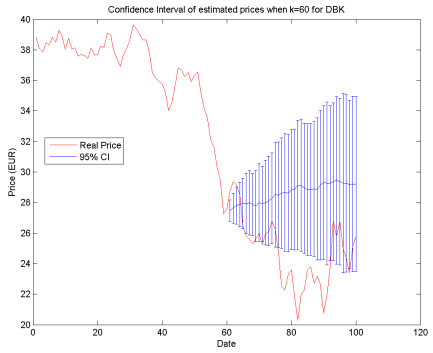
component case. The confidence interval plots are presented in Figure 3.37. Since the closing prices for component III are relatively much smaller, the prices have different scale as the other two components to make it possible to see the plot. For all 3 cases, all of the closing prices lie within the confidence interval. This is a perfect result. Especially for component LGEN, the average variance has almost the same trend as the trend of real prices in "future". Cosine similarity is applied as the distance measurement in the experiment. In Figure 3.39, the variances of log–return factors of predicted "future" part are plotted against time. Approximately 2 clusters formed with the first cluster contains relatively higher variances and there are only 3 components within. The second cluster contains the rest of components. The general trend of variances is a linear upward trend but there is a sharp peak at around date 13 and the average variance is positioned between the first and second cluster. Let us choose component RBS for "high variance" case, component HSBA for "average variance" case and component III for "small



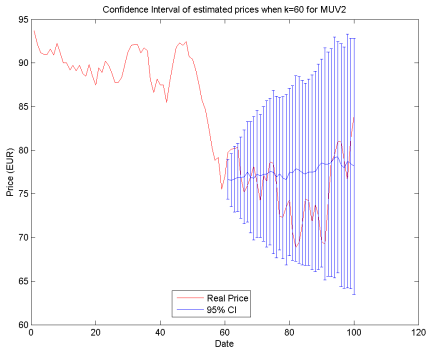
(A) Individual Experiment



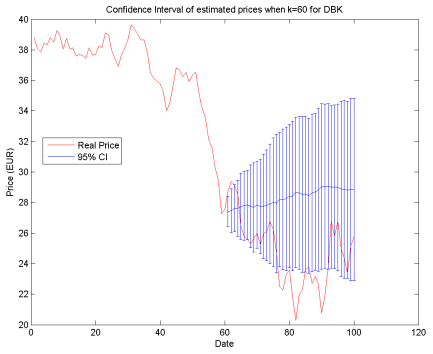
(B) Individual Experiment



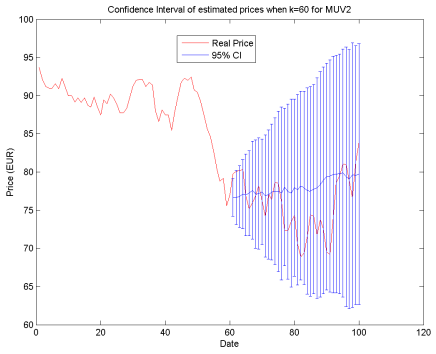
(C) Whole Experiment



(D) Whole Experiment



(E) Financial Sectors Experiment



(F) Financial Sectors Experiment

FIGURE 3.32: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 40 for component DBK and MUV2 (Cosine Similarity) for individual experiment, whole experiment and financial sectors experiment. Variance of predicted log–return factors against time for these components plot is shown in the end.

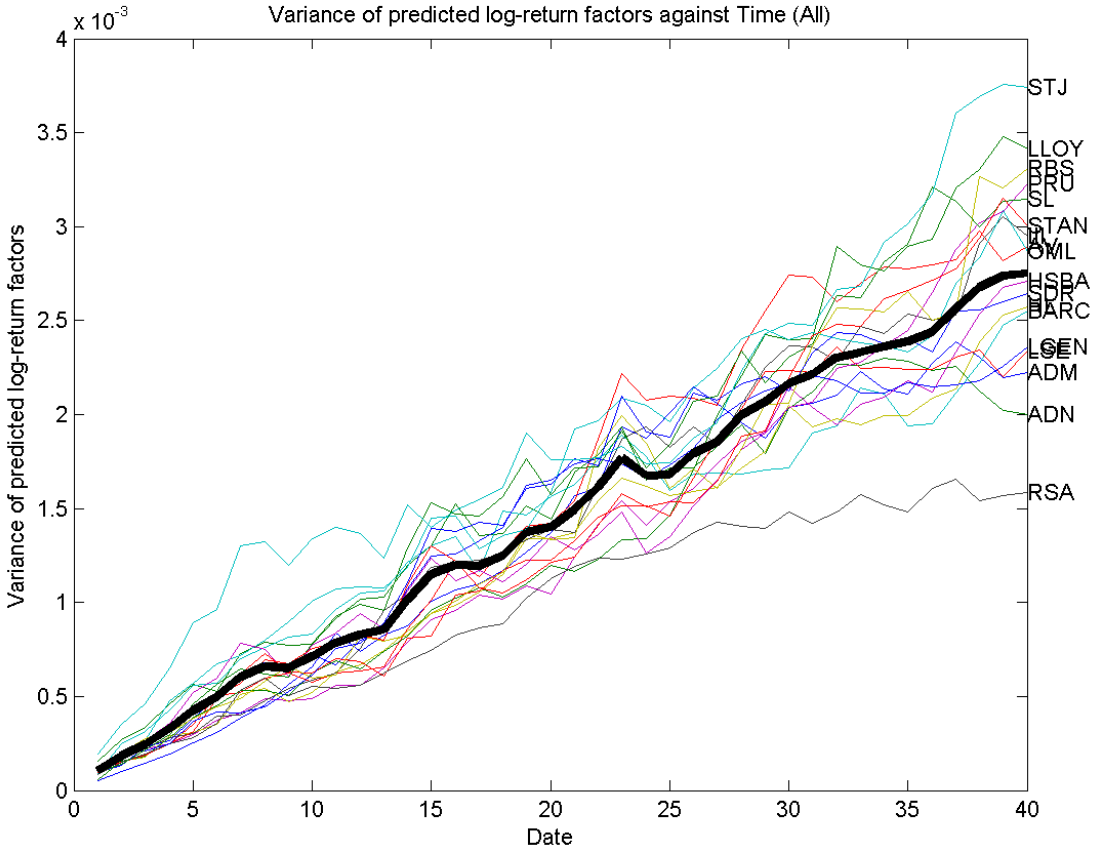
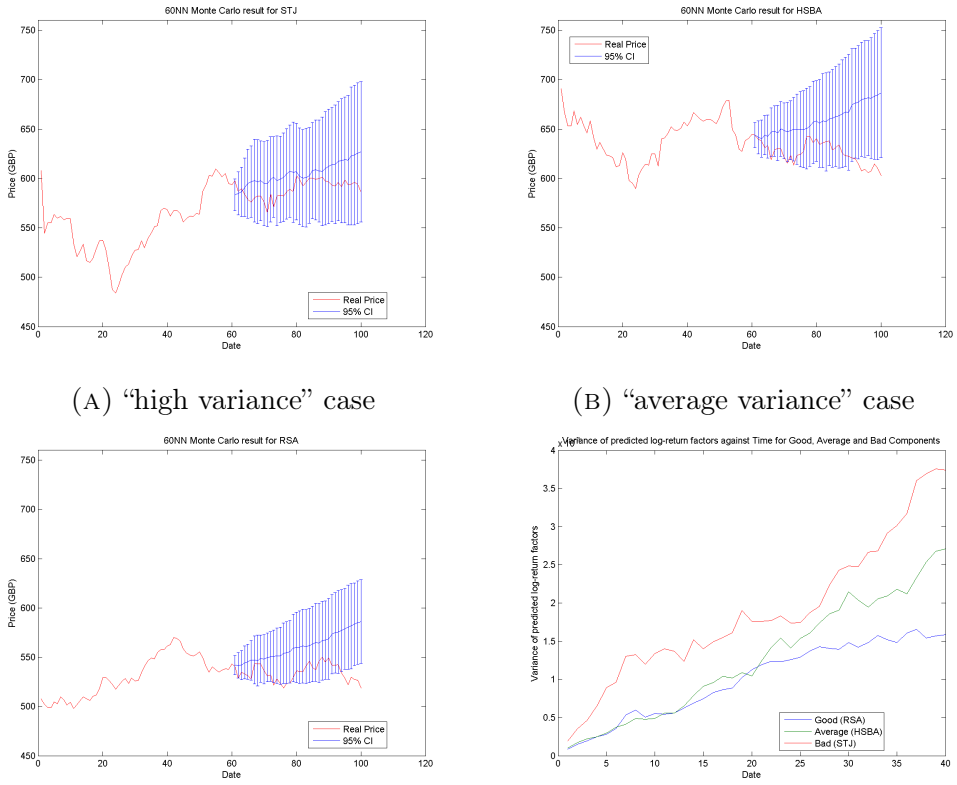


FIGURE 3.33: Graph of variance of predicted log–return factors against time for selected components of financial sectors of FTSE100 index when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (Euclidean Distance). The name of component is presented on the last variance and the thick black line represents the average variance of these factors of selected components.

variance" case. From Figure 3.40, it is observed that the results for all 3 cases are good since all real prices lies within the confidence interval. For "high variance" and "average variance" cases, the predicted mean prices are closed to the real price which means this experiment has good performance for these 2 components. The results of the experiment applying some distance measurement are quite good. In general, the results using correlation distance and cosine similarity are better than the results using Euclidean distance and City Block distance while the variances of log–return factors are approximately 3 times larger. Hence it could be concluded that for data from closing prices of components of FTSE100, the results using correlation distance and cosine similarity are better since the confidence interval could have good results.



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) Variance Plot

FIGURE 3.34: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 40 for selected "high variance", "average variance" and "high variance" components (Euclidean Distance) for financial sectors experiment of FTSE100 index. Variance of predicted log–return factors against time for these components plot is shown in the end.

3.4 The comparison between the ARMA model forecasting with the kNN experiment

Autoregressive moving average model (or the ARMA model) is mainly studied in time series forecast problem. Comparison between the result of the ARMA model and the kNN experiment is mainly studied in this section. For simplicity, $p = 1$ and $q = 1$ are chosen as parameters for the ARMA model. The kNN experiment, $k = 30$, "present" length is 60, "future" length is 40. Figure 3.41 is the plot of DAX index closing prices against time and the maximum "history" part, "present" part and "future" part is shown in different colors. It is clear that the DAX index closing prices have generally linear upward trend over the whole "history" part. Then in the middle of "present" part, there is a sharp drop in the closing price. A crisis can be detected for such case. The closing prices reach the bottom in the

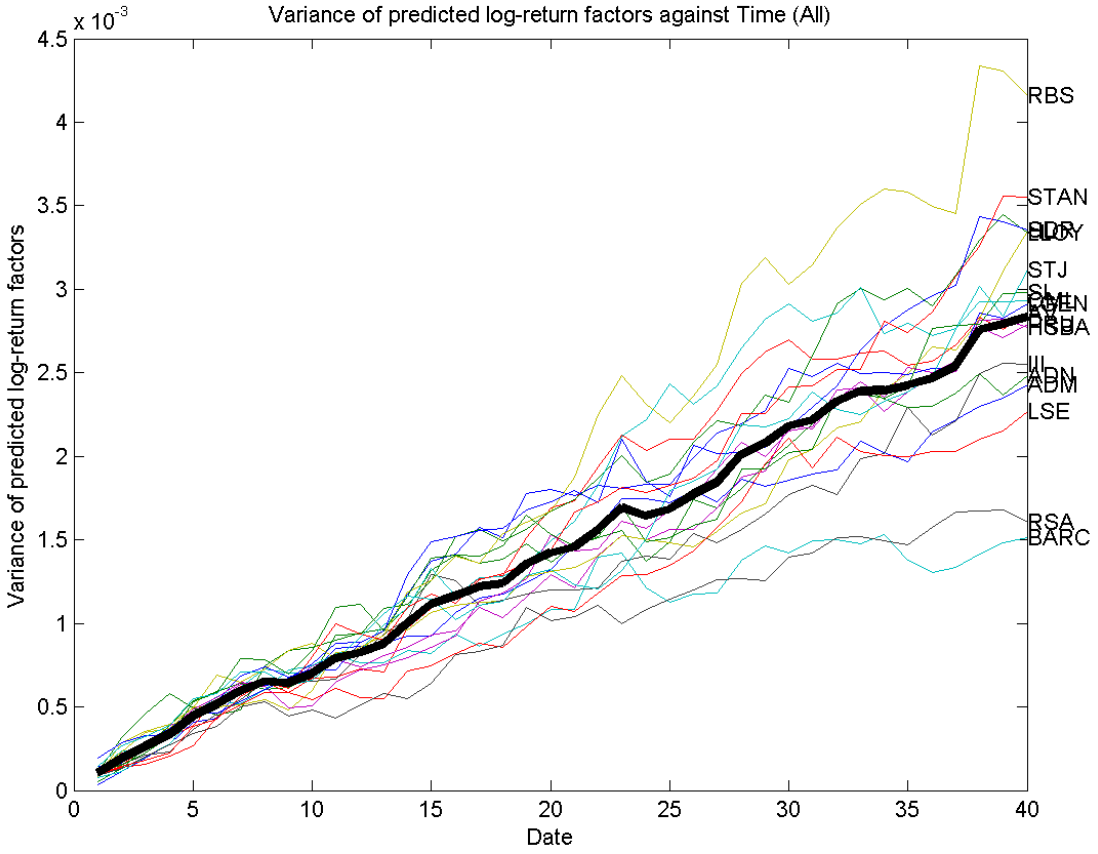
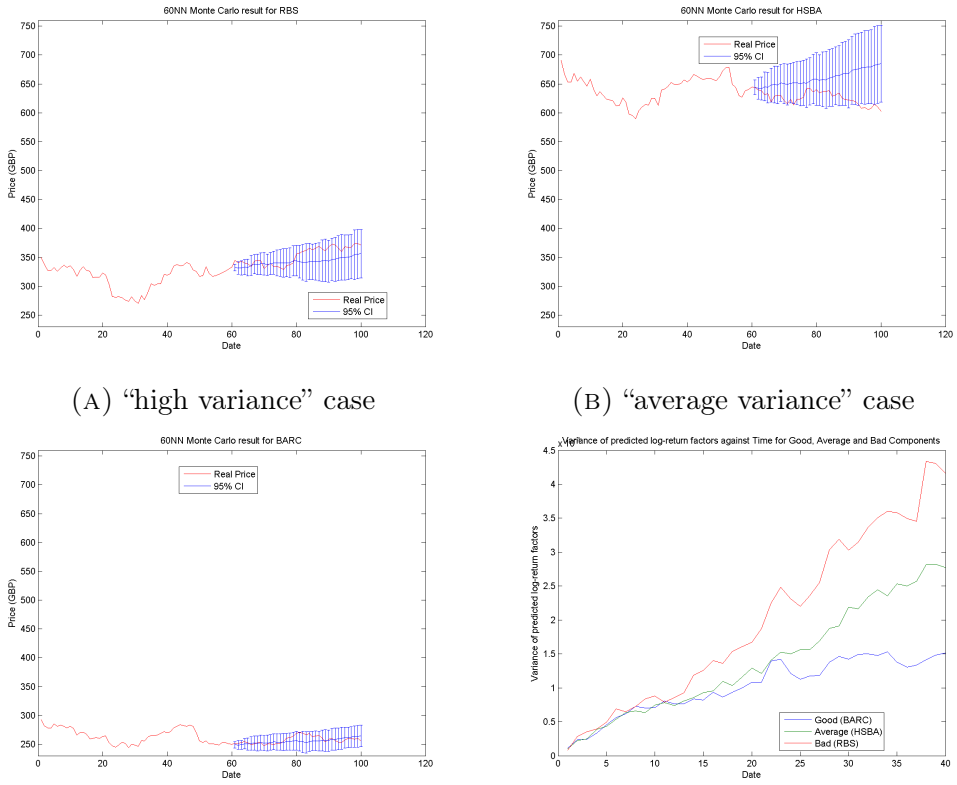


FIGURE 3.35: Graph of variance of predicted log–return factors against time for selected components of financial sectors of FTSE100 index when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (City-block Distance). The name of component is presented on the last variance and the thick black line represents the average variance of these factors of selected components.

beginning of "future" part and then it has a general horizontal trend. 4 different lengths of "history" part are set. The total length of whole"history" part is 355 and 4 different lengths represents 25%, 50%, 75% and 100% of the whole "history" part. 4 measurements of distances are chosen as similar as in the previous experiment. The results of comparisons between the ARMA(1,1) and the kNN experiment are represented as figures of time series. In the result figures, the blue time series is the closing prices of the component over a specific time interval. The whole time series is split by red lines into 3 parts, "history" part with 4 different lengths, "present" part and "future" part. Two components are selected from the list of components of DAX index as "Good" and "Bad" results for our prediction. For each component, 4 figures are generated as result of comparison. Euclidean distance is applied as the distance measure for the kNN experiment. Figure 3.42 represents the forecast closing price for the ARMA(1,1) model and average closing prices of



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) Variance Plot

FIGURE 3.36: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 40 for selected "high variance", "average variance" and "high variance" components (City Block Distance) for financial sectors experiment of FTSE100 index. Variance of predicted log–return factors against time for these components plot is shown in the end.

nearest neighbors generated from the kNN experiments, the 95% confidence intervals are plotted in the same plot. All sub–Figures 3.42a3.42c3.42e3.42g show the comparison result for component FME. It is clear to show that both models give good approximation for "future" part. These results are quite similar since the 95% confidence intervals are almost the same. However, when the length of "history" part is increased to 100%, the average predicted closing price from the kNN experiment gives a better prediction result since it successfully predict an upward trend when the real prices go up in "future" part. The results of component FME is positive. The sub–Figures 3.42b3.42d3.42f3.42h show the comparison result for component DB1. In general, these sub–figures show that the predicted results are not accurate. The ARMA(1,1) forecast result and kNN average predicted closing prices have completed different trend as the real closing prices. The reason may because there is a sharp drop at the end of "present" part. Both methods are limited to predictability by using a history of "normal" prices to predict the

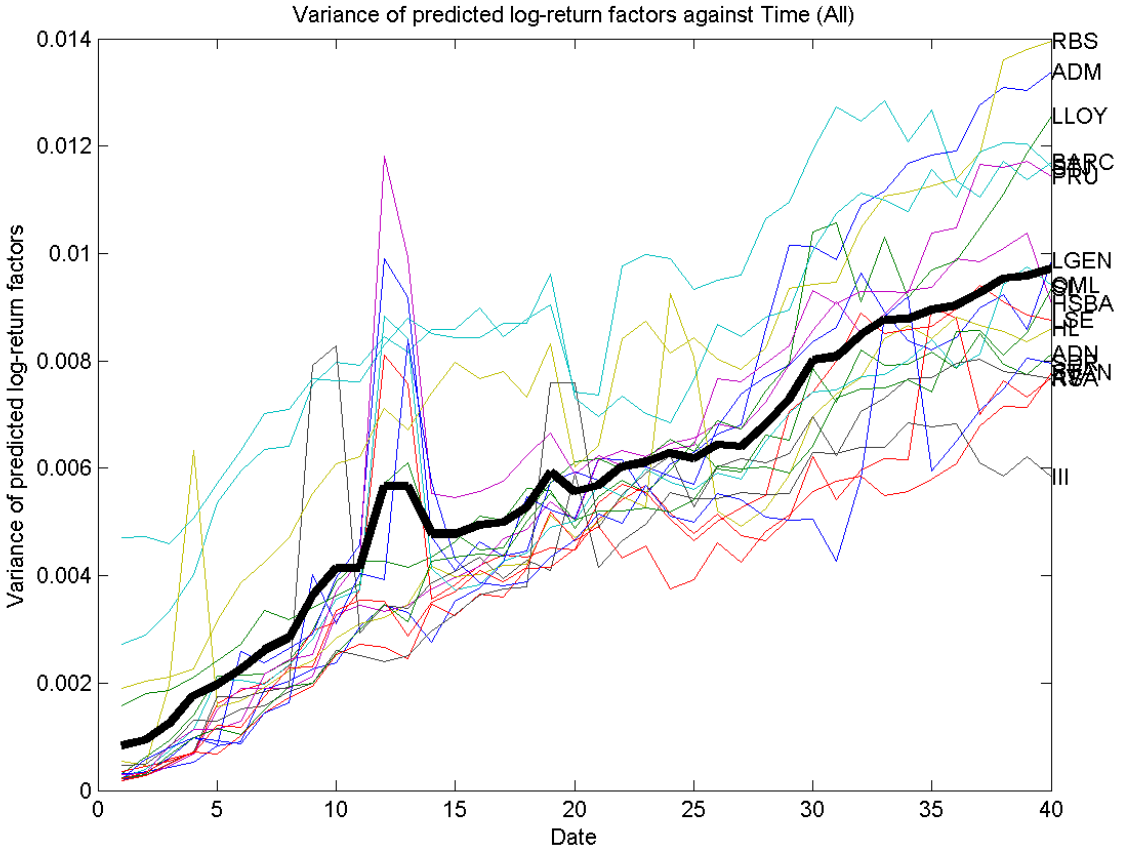
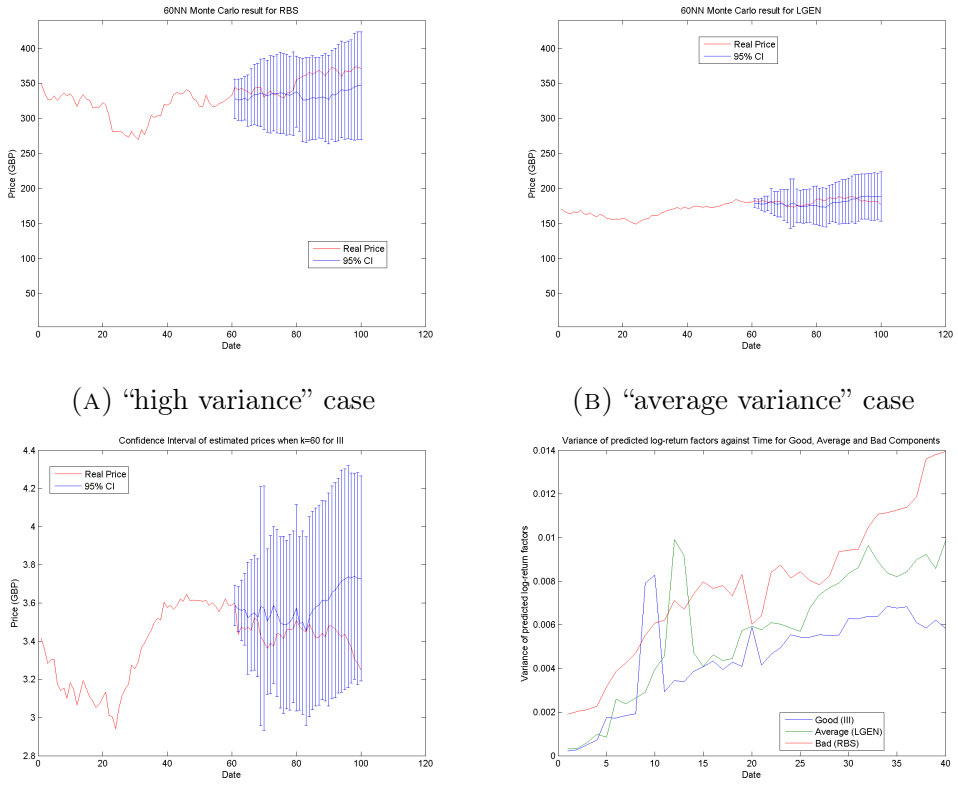


FIGURE 3.37: Graph of variance of predicted log–return factors against time for selected components of financial sectors of FTSE100 index when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (Correlation Distance). The name of component is presented on the last variance and the thick black line represents the average variance of these factors of selected components.

crisis. Cityblock distance is applied for the kNN experiment and then the result is compared with the ARMA(1,1) model forecast closing prices. Figure 3.43 list the presents the results for component FME and component DB1. The Subfigures 3.43a 3.43c 3.43e 3.43g of results for component FME gives a good forecast for both models. However, when "history" part is longer, the predicted average prices from the kNN experiment has a better prediction than the ARMA(1,1) model since the ARMA(1,1) model gives a horizontal trend when "history" part is longer. For component DB1, Subfigures 3.43b 3.43d 3.43f 3.43h show that both models have relatively bad performance. For this component, the ARMA(1,1) model gives an upward trend while the real prices are having a downward trend. Almost all real prices lie outside the confidence interval. Although the kNN experiment gives an upward trend as well, but the average predicted prices are closer to the real



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) Variance Plot

FIGURE 3.38: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 40 for selected "high variance", "average variance" and "high variance" components (Correlation Distance) for financial sectors experiment of FTSE100 index. Variance of predicted log–return factors against time for these components plot is shown in the end.

closing price than the forecast results from the ARMA(1,1) model. For both components, it is obvious that the confidence intervals for both models are similar and the prediction results are having similar trend. Correlation distance is applied as distance measurement for kNN method. Figure 3.44 represents the comparisons between the kNN experiment and the ARMA(1,1) model for component FME and DB1. From Subfigures 3.44a 3.44c 3.44e 3.44g, prediction results of two models for component FME is relatively good since the average prediction prices are very close to the real closing prices. It is also shown that, when "history" part is longer, the kNN experiment improves its performance of prediction while the ARMA(1,1) gives a worse prediction when "history" part is longer. However, for component DB1, the Subfigures 3.44b 3.44d 3.44f 3.44h show that the prediction results for both methods are quite bad. They failed to predict the trends of the real closing prices since the predicted prices have upward trend when the real closing prices have downward trend. The prediction results of two methods are quite similar and

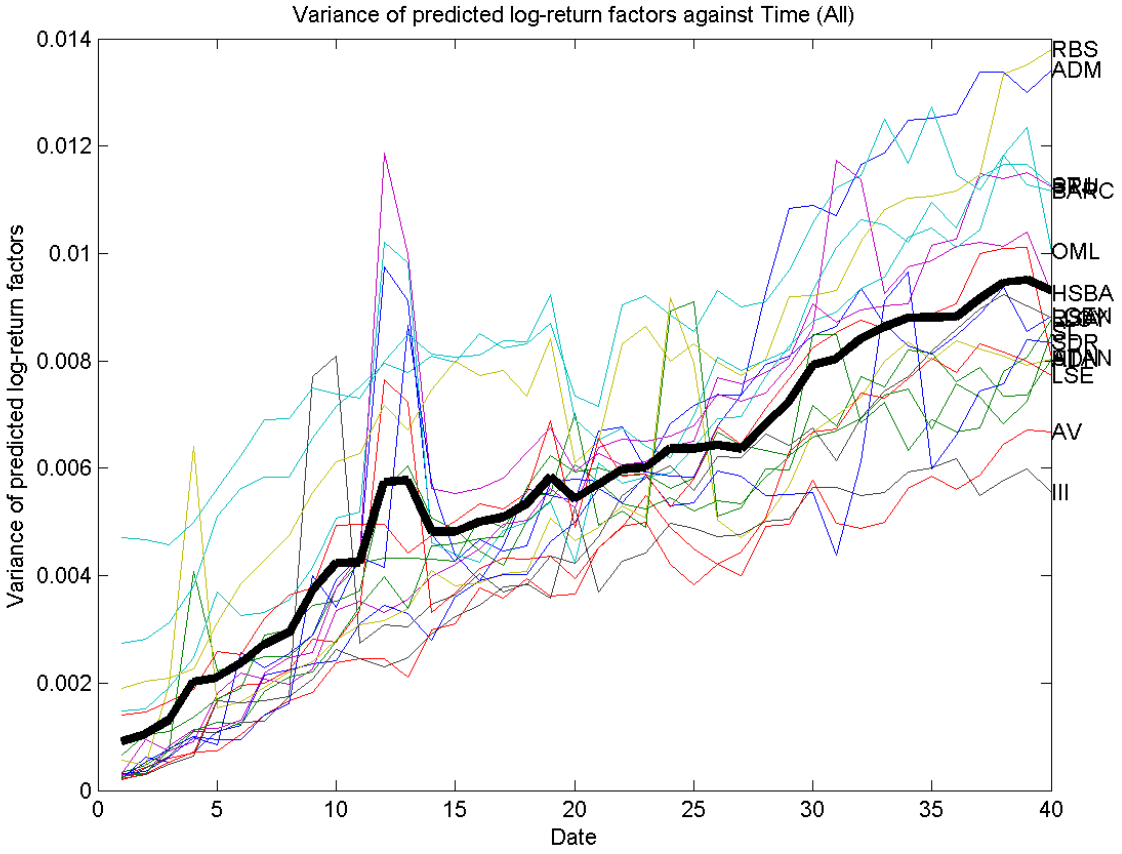
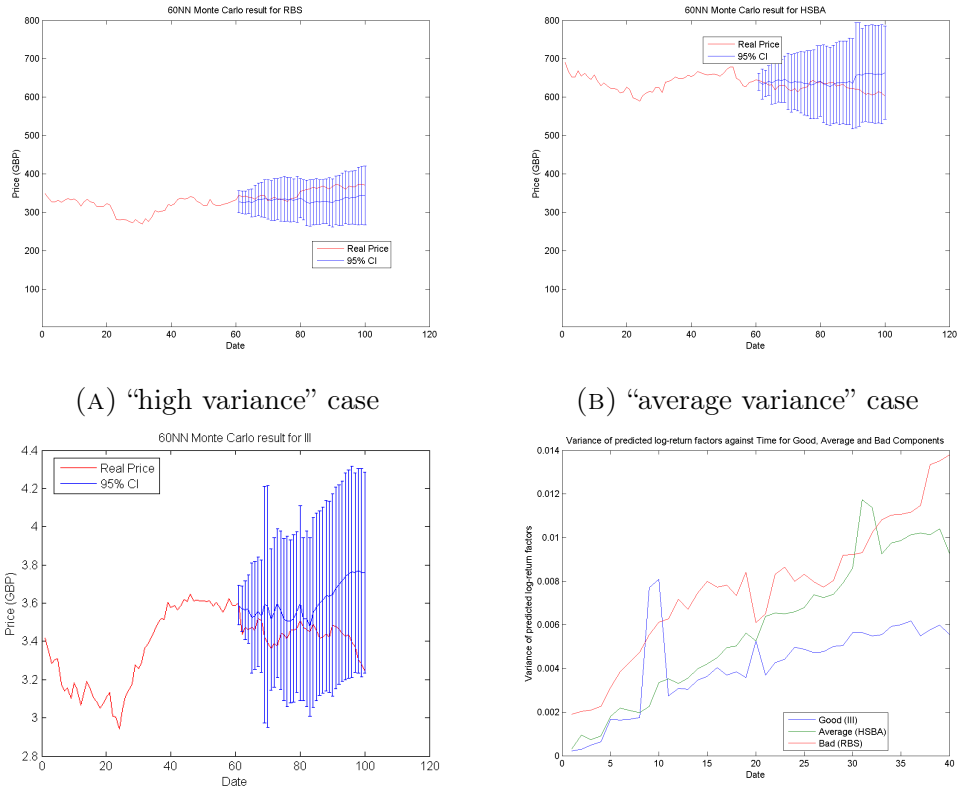


FIGURE 3.39: Graph of variance of predicted log–return factors against time for selected components of financial sectors of FTSE100 index when $k = 60$, "present" fragment has length 60 and "future" fragment has length 40 (Cosine Similarity). The name of component is presented on the last variance and the thick black line represents the average variance of these factors of selected components.

the confidence intervals are almost the same except for several specific length of "history" part, the confidence interval is relatively wide in the end. Cosine similarity is then applied as distance measurement for the kNN experiment. Figure 3.45 represents the comparison between prediction results of the kNN experiment and the ARMA(1,1) model. For component FME, Subfigures 3.45a 3.45c 3.45e 3.45g show that average predicted closing prices from the kNN experiment have similar trend as the trend of real closing prices and the longer the "history" part is, the better quality of the prediction. The ARMA(1,1) model has slightly worse prediction results than the kNN experiment since when real prices have an upward trend, the ARMA(1,1) model predicted prices have a horizontal trend. The Subfigures 3.45b 3.45d 3.45f 3.45h represents the comparison results for component DB1. These results give bad prediction since the predicted prices have upward trend while the real prices have downward trend. Almost all real prices lie outside



(A) "high variance" case

(B) "average variance" case

(C) "small variance" case

(D) Variance Plot

FIGURE 3.40: 95% Confidence interval of predicted price and real price plots of "present" and "future" for $k = 60$, present fragment length 60, future fragment length 40 for selected "high variance", "average variance" and "high variance" components (Cosine Similarity) for financial sectors experiment of FTSE100 index. Variance of predicted log–return factors against time for these components plot is shown in the end.

the 95% confidence intervals. Hence for this component, both models failed to predict the "future" part closing prices. By looking at the results of comparisons for selected components with 4 distance measurements for the kNN experiment, it is clear that for all various distance types, for component FME, an example of "good" results, both the kNN experiment and the ARMA(1,1) model have good prediction and successfully predict the trend of the real closing prices. Almost all real prices lie within the 95% confidence interval and the average predicted prices of the kNN experiment and the ARMA(1,1) predicted prices are very close. For component DB1, an example of "bad" results, both the kNN experiment and the ARMA(1,1) have bad prediction and they cannot predict the closing prices of "future" part. Almost all real prices lie outside the 95% confidence interval. By meaning of changing the distance measurement, experiment using Euclidean distance and City block distance have similar results. The average predicted prices have

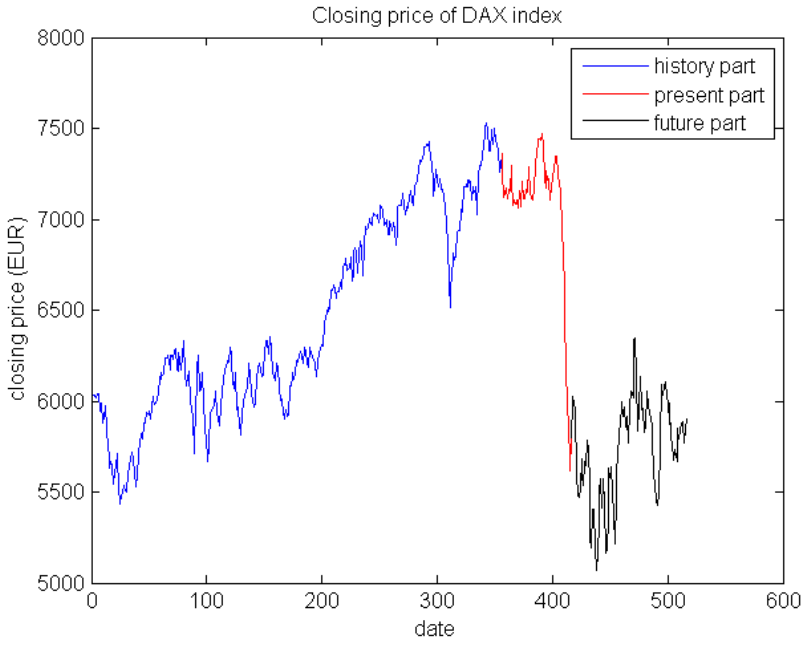
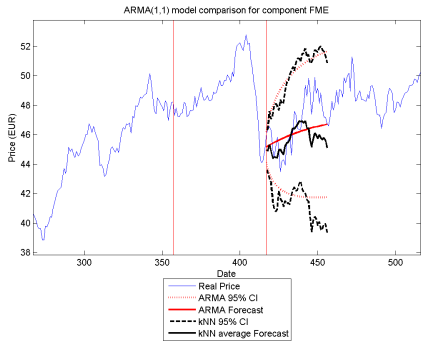


FIGURE 3.41: Graph of DAX index closing price against time for the ARMA(1,1) model comparison experiment

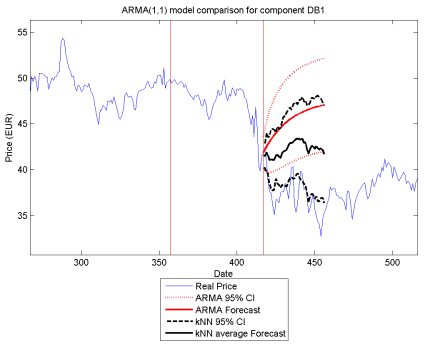
slightly worse performance but the confidence interval is relatively narrower. Experiment using Correlation distance and Cosine similarity, the average predicted prices have slightly better performance but the confidence interval is relatively wider. Overall, by looking at comparison results for all 30 components, the results of the kNN experiment are similar to the ARMA(1,1) model. Hence it can be concluded that for these components, the kNN experiment could predict some future closing prices and therefore this experiment has checked that the history contains some information about future closing prices.

3.5 The application of the Lorenz attractor in the kNN experiment

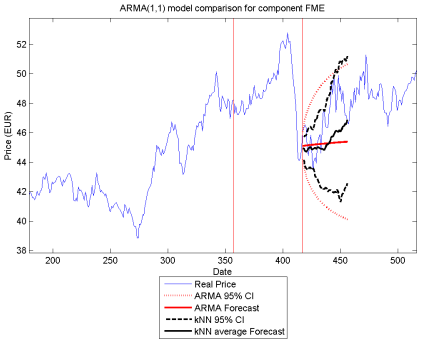
Random initial conditions are used to solve the system of differential equations and the solutions of the Lorenz system are computed for $t \in [0, 1000]$ with step size 0.5. In Figure 3.46, the solutions of the Lorenz attractor with random initial conditions for $t \in [0, 1000]$ with step size 0.5 are computed. These figures have general similar shape to our example in Figure 3.4. The figures are less smooth since we set a larger step size. Figure 3.47 are the solutions of the Lorenz attractor against time. There exists oscillation phenomenon for all three solutions and no



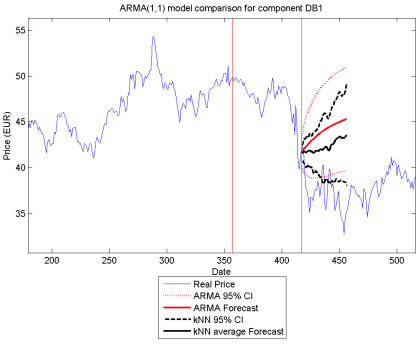
(A) 25% "history" FME



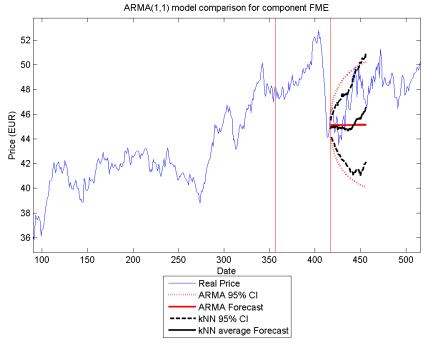
(B) 25% "history" DB1



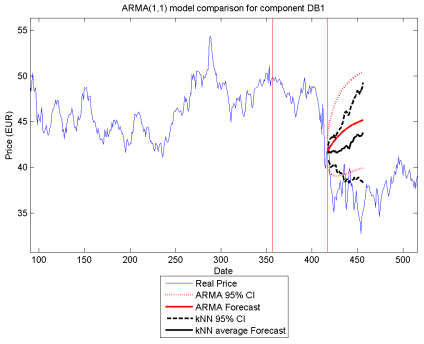
(C) 50% "history" FME



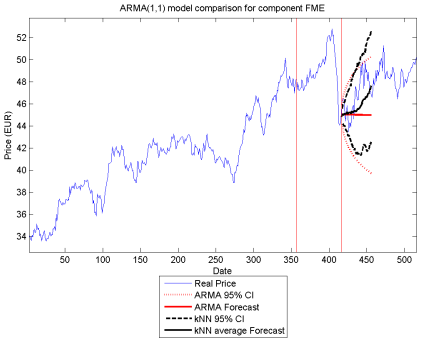
(D) 50% "history" DB1



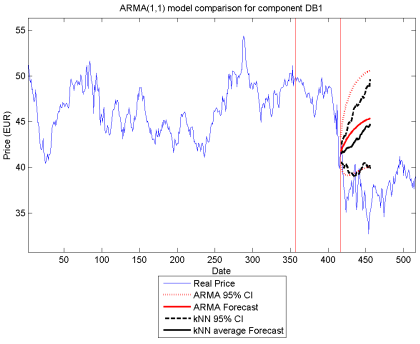
(E) 75% "history" FME



(F) 75% "history" DB1

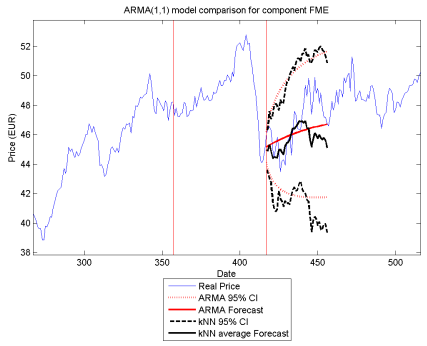


(G) 100% "history" FME

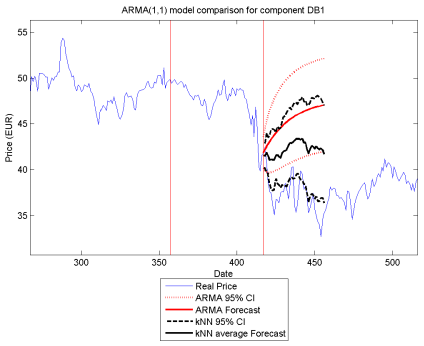


(H) 100% "history" DB1

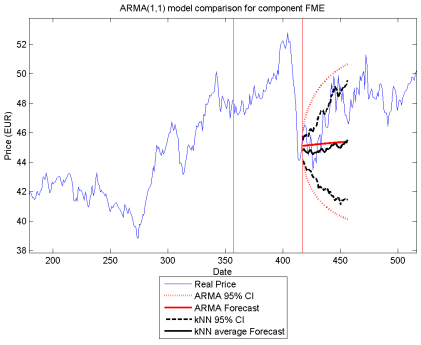
FIGURE 3.42: 95% Confidence interval comparison between the ARMA(1,1) and the kNN experiment (Euclidean Distance) for component FME (all LEFT figures) and DB1 (all RIGHT figures)



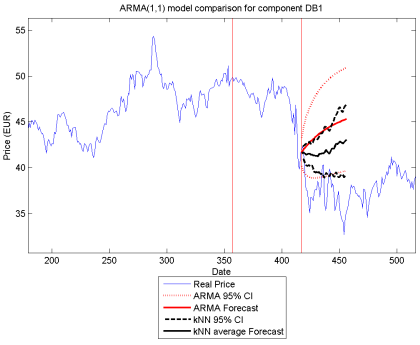
(A) 25% "history" FME



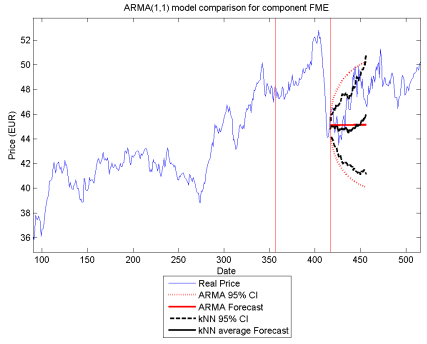
(B) 25% "history" DB1



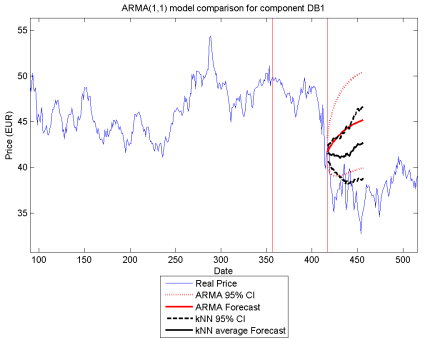
(C) 50% "history" FME



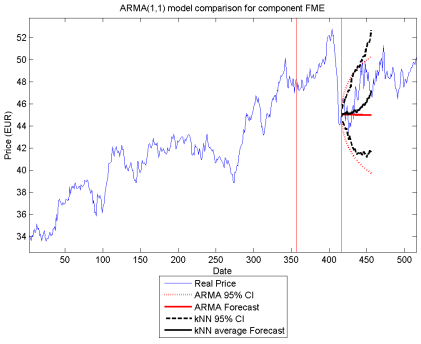
(D) 50% "history" DB1



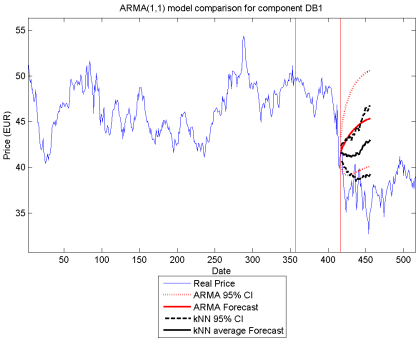
(E) 75% "history" FME



(F) 75% "history" DB1

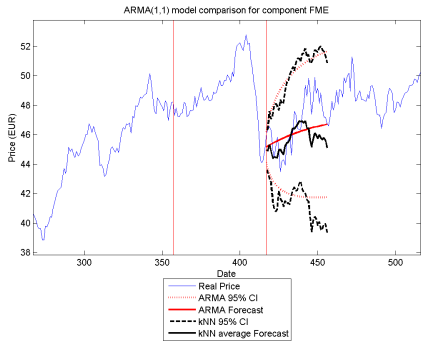


(G) 100% "history" FME

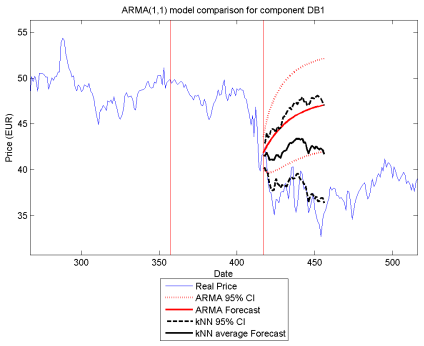


(H) 100% "history" DB1

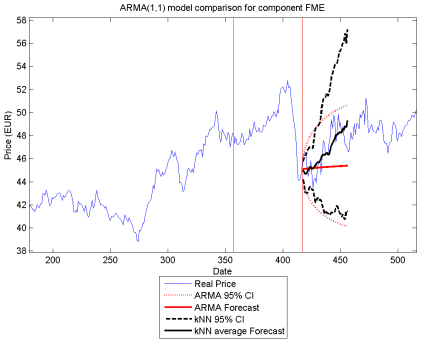
FIGURE 3.43: 95% Confidence interval comparison between the ARMA(1,1) and the kNN experiment (City block Distance) for component FME (all LEFT figures) and DB1 (all RIGHT figures)



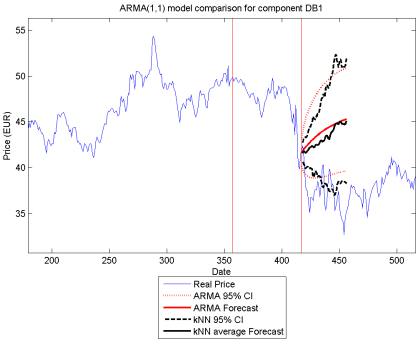
(A) 25% "history" FME



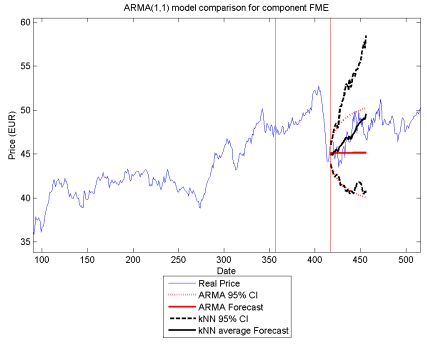
(B) 25% "history" DB1



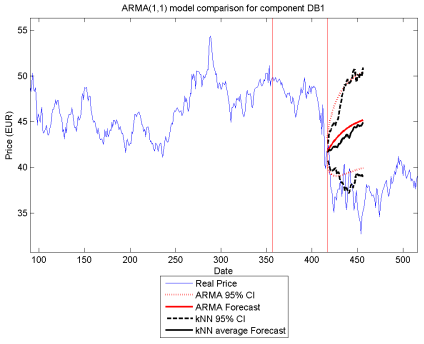
(C) 50% "history" FME



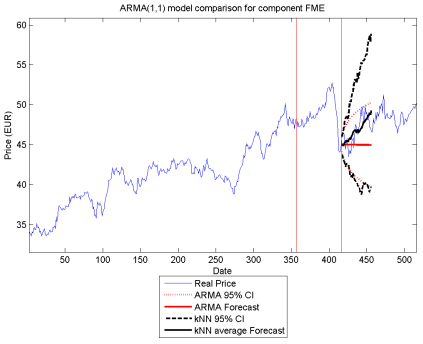
(D) 50% "history" DB1



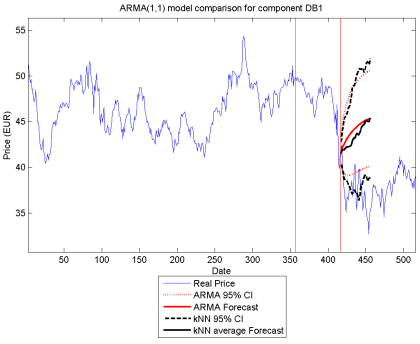
(E) 75% "history" FME



(F) 75% "history" DB1

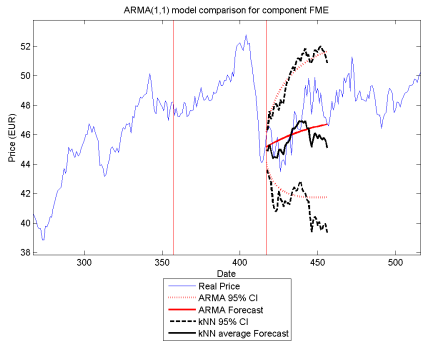


(G) 100% "history" FME

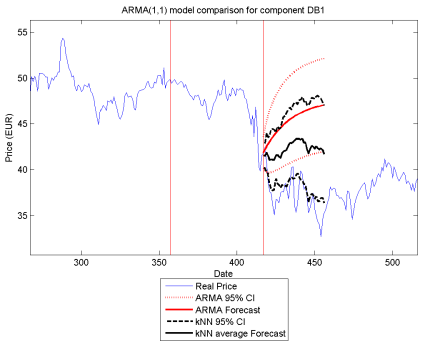


(H) 100% "history" DB1

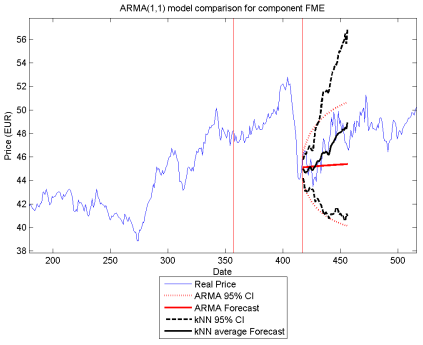
FIGURE 3.44: 95% Confidence interval comparison between the ARMA(1,1) and the kNN experiment (Correlation Distance) for component FME (all LEFT figures) and DB1 (all RIGHT figures)



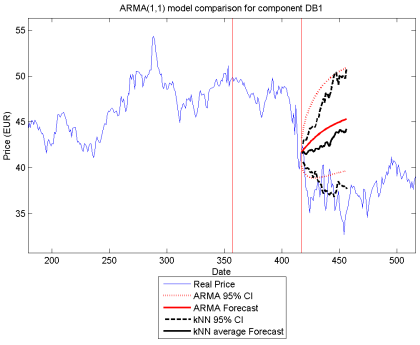
(A) 25% "history" FME



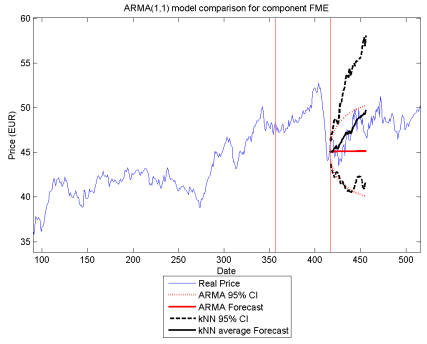
(B) 25% "history" DB1



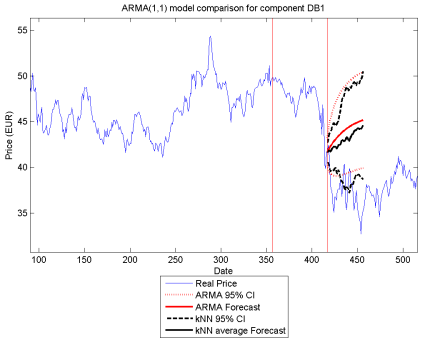
(C) 50% "history" FME



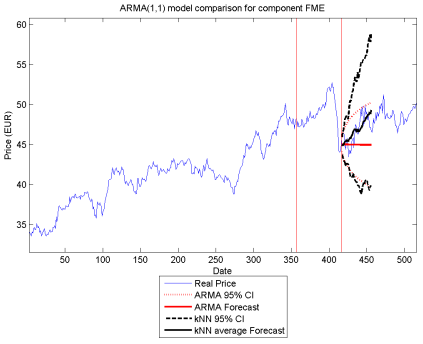
(D) 50% "history" DB1



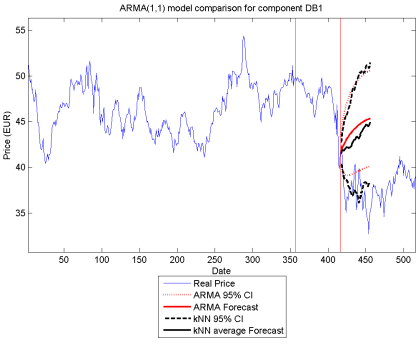
(E) 75% "history" FME



(F) 75% "history" DB1

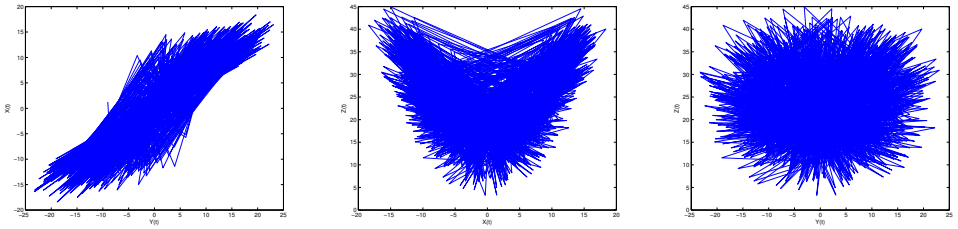


(G) 100% "history" FME



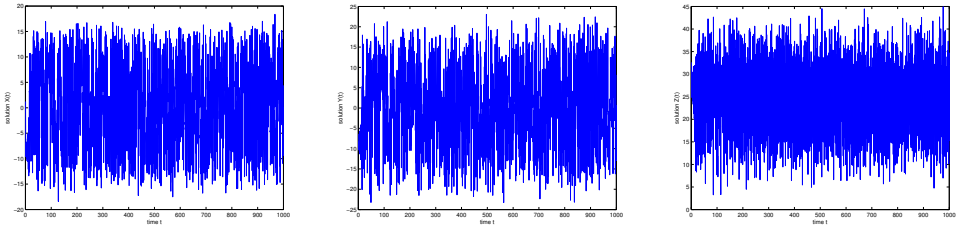
(H) 100% "history" DB1

FIGURE 3.45: 95% Confidence interval comparison between the ARMA(1,1) and the kNN experiment (Cosine Similarity) for component FME (all LEFT figures) and DB1 (all RIGHT figures)



(A) $X(t)$ against $Y(t)$ (B) $Z(t)$ against $X(t)$ (C) $Z(t)$ against $Y(t)$

FIGURE 3.46: The Lorenz attractor with random initial conditions for $t \in [0, 1000]$ with step size 0.5.



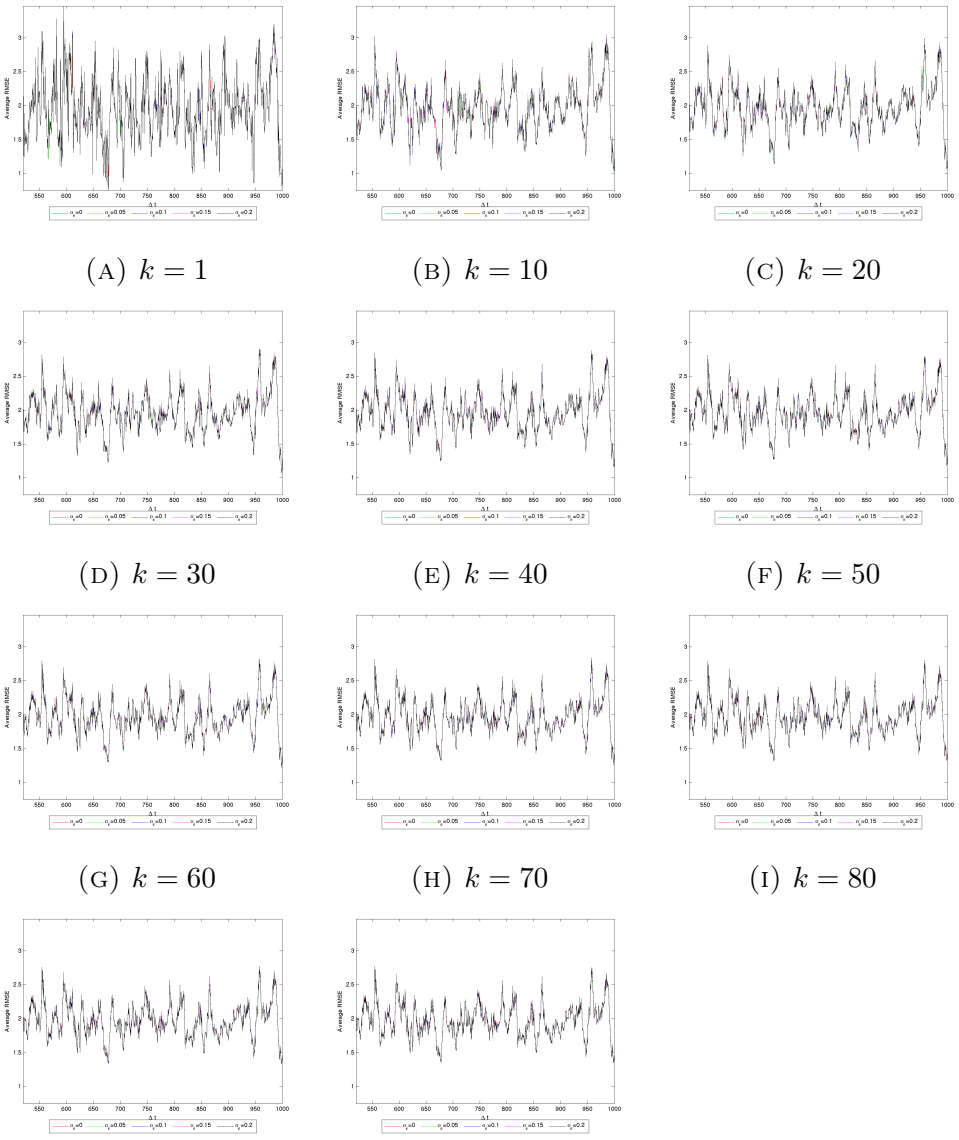
(A) $X(t)$ against time (B) $Y(t)$ against time (C) $Z(t)$ against time

FIGURE 3.47: The Lorenz attractor against time with random initial conditions for $t \in [0, 1000]$ with step size 0.5.

clear trend is discovered. The solution $X(t)$ is roughly oscillating between -15 and 15; the solution $Y(t)$ is roughly oscillating between -20 and 20; the solution $Z(t)$ is roughly oscillating between 10 and 40. Among these 3 solutions, $Z(t)$ has better trend than the other two solutions. Hence, the prediction result for $Z(t)$ is expected to be better than the other two solutions. To visualize the change in the results for different k and α_ϵ, plots of average RMSE and standard deviations are computed. The level of noise α_ϵ is chosen between 0 and 0.2 with step size 0.05. In this section, we use average RMSE against time steps plot and average standard deviation against 'future' time plot to visualize the kNN experiment result. For different k and α_ϵ, we would like to discuss the changes in kNN results.

3.5.0.1 The average relative means square error

RMSE measures the relative the kNN experiment prediction accuacy for every 'window' in our test set (i.e. there are 960 'windows' prediction in test set, $(21, 22, \ldots, 40), (22, 23, \ldots, 41), \ldots, (981, 982, \ldots, 1000)$). Average RMSE measures the average prediction accuacy and this gives us information on how the average changes for different k (i.e. $k = 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100$). Figure 3.48, Figure 3.49 and Figure 3.50 are average RMSE plots against time



(A) $k = 1$ (B) $k = 10$ (C) $k = 20$

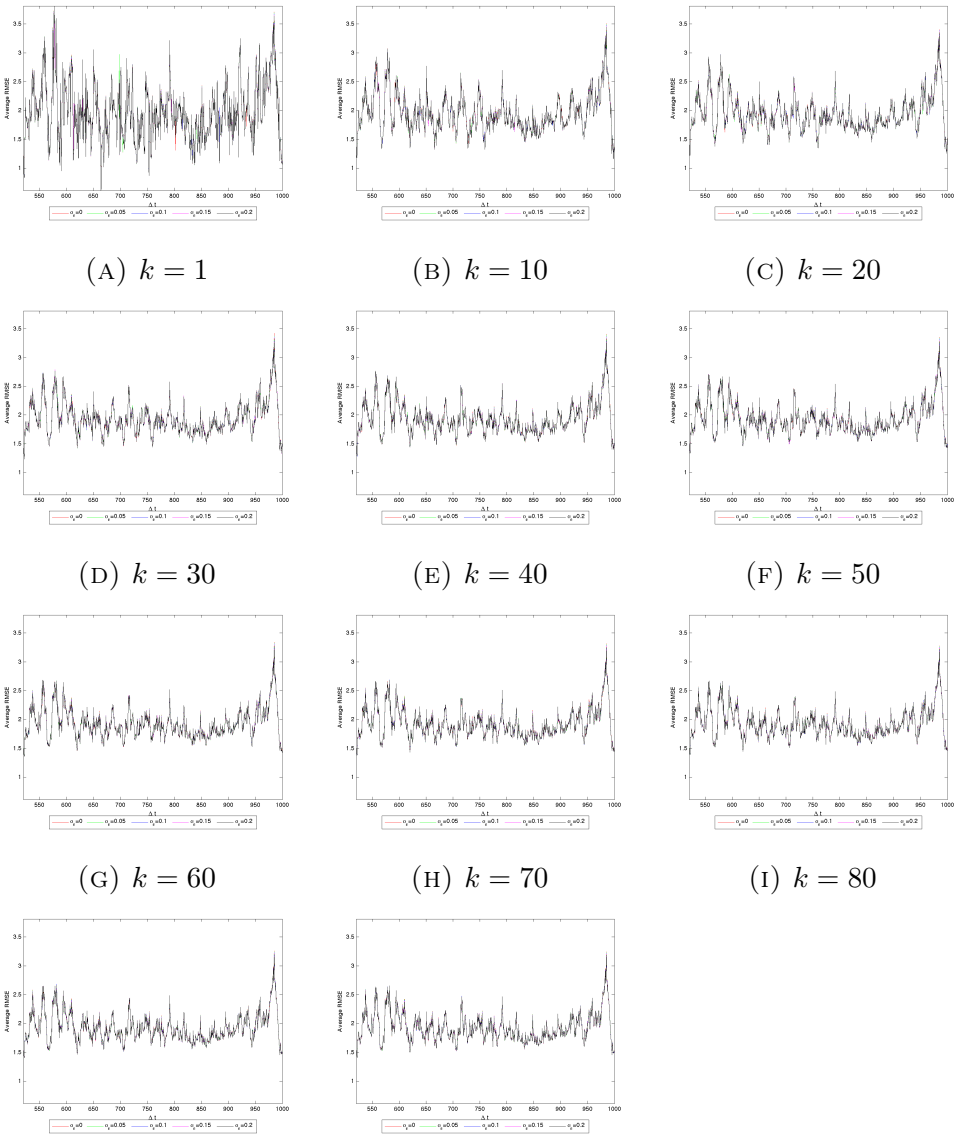
(D) $k = 30$ (E) $k = 40$ (F) $k = 50$

(G) $k = 60$ (H) $k = 70$ (I) $k = 80$

(J) $k = 90$ (K) $k = 100$

FIGURE 3.48: Average RMSE of all k predictions against Δt (change in time step) for solution $X(t)$.

step for solution $X(t)$, $Y(t)$ and $Z(t)$. In general, the average RMSE plots of each solution for different α_ϵ are almost identical. When α_ϵ is higher, the amplitude of average RMSE is higher. Comparing with the average RMSE across all solution, it is clear that the average RMSEs for solution $X(t)$ and $Y(t)$ are similar and they are much higher than the average RMSE for solution $Z(t)$ (i.e. all average RMSE for $Z(t)$ are less than 0.45). The average RMSE for all solutions are having oscillating trend. For plots when $k = 1$, the average RMSE plots have larger peaks for some values. In other words, the kNN experiment results can be very good or very bad when $k = 1$. When k increases to 100, the average RMSE plots have smaller peaks but the trend is similar to the average



(A) $k = 1$ (B) $k = 10$ (C) $k = 20$

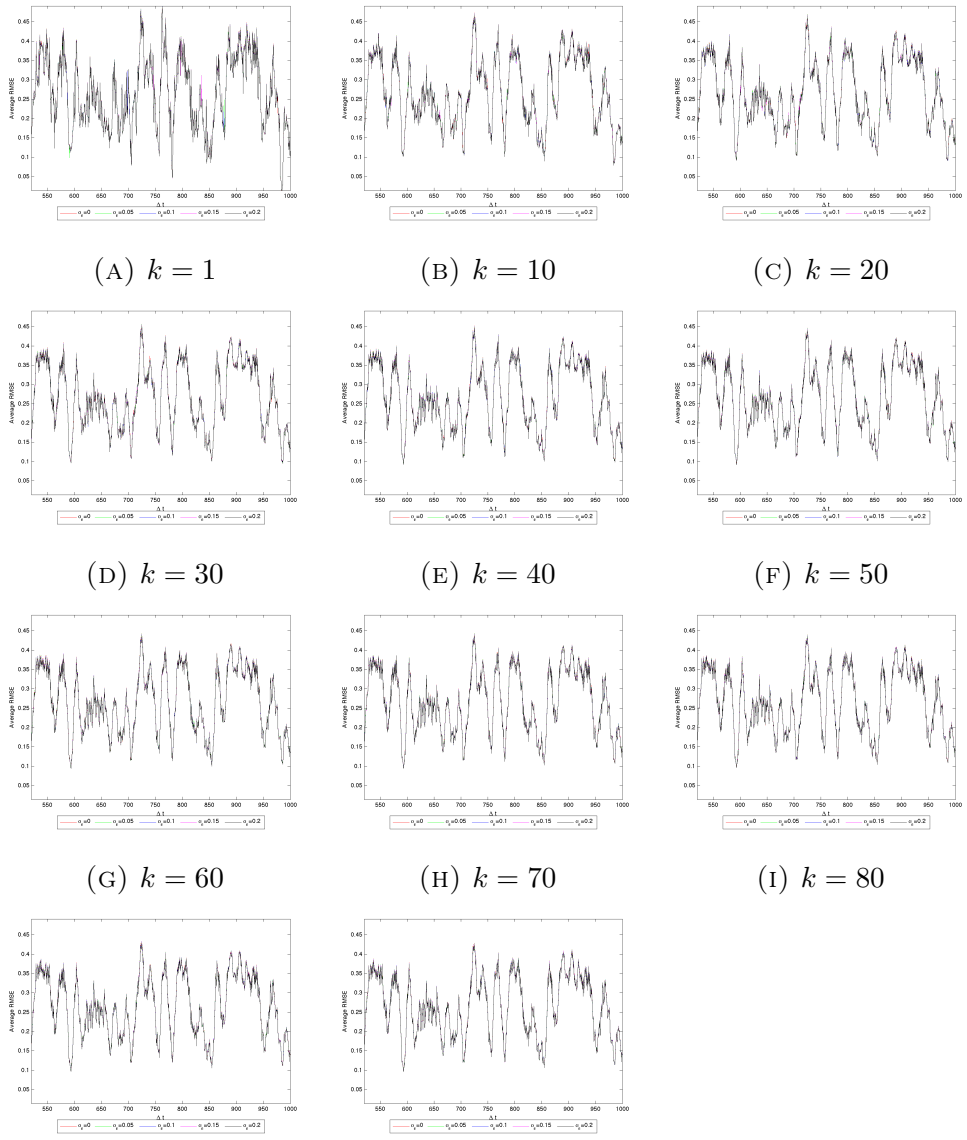
(D) $k = 30$ (E) $k = 40$ (F) $k = 50$

(G) $k = 60$ (H) $k = 70$ (I) $k = 80$

(J) $k = 90$ (K) $k = 100$

FIGURE 3.49: Average RMSE of all k predictions against Δt (change in time step) for solution $Y(t)$.

RMSE when $k = 1$. The distribution of average RMSE for all k for all solutions are represented in Figure 3.51. It is easy to see that the mean values of average RMSE for each solution for all k are similar. But the tails of boxplots are converge to some values when $k \to 100$. For $X(t)$ and $Y(t)$, the values Q_1 and Q_3 converge as k increases while for $Z(t)$, the values Q_1 and Q_3 are generally similar.



(A) $k = 1$ (B) $k = 10$ (C) $k = 20$

(D) $k = 30$ (E) $k = 40$ (F) $k = 50$

(G) $k = 60$ (H) $k = 70$ (I) $k = 80$

(J) $k = 90$ (K) $k = 100$

FIGURE 3.50: Average RMSE of all k predictions against Δt (change in time step) for solution $Z(t)$.

3.5.0.2 The average standard deviation

Standard deviations calculated in this experiment measure the spread of prediction at a specific day for all 'window'. The average standard deviation is the mean of k standard deviations. Figure 3.52, Figure 3.53 and Figure 3.54 contain the standard deviations of real 'future' and average standard deviations of k predicted future. By comparing the standard deviation of real 'future' with the average standard deviation of k predicted future, it seems that solution $X(t)$ has the most similar standard deviation. For solution $X(t)$, when $k = 1$ we have an oscillation of average standard deviation with a horizontal trend. The oscillation amplitude is

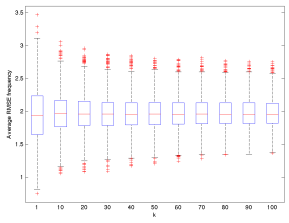
slightly reduced as $k \to 100$. It is interesting to see that there is a sharp increase at date 20. For solution $Y(t)$, the predicted average standard deviations are generally smaller than the standard deviations of real 'future'. When $k = 1$, the average standard deviations have a generally decreasing trend and at date 20, there is a sharp peak. As $k \to 100$, the average standard deviations have horizontal trend and they have oscillations and the amplitudes are similar. For solution $Z(t)$, the average standard deviation predictions are generally higher than the standard deviation of real 'future'. For $k = 1$, there is an uptrend and at date 20, a peak forms. When $k \to 100$, the average standard deviations converge to a parabola with a high peak in the end.

3.6 Conclusion

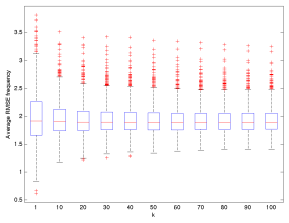
A new approach of time series predictability method using kNN algorithm is formulated and different experiments have been performed based on this idea. For "present" time frame daily log–returns, k nearest neighbors are searched by applying kNN algorithm from a set of "history" time frame. These nearest neighbors are treated as the log–returns for "future". This approach also uses ideas of dynamical system to reconstruct the "future" closing prices from these nearest neighbors. For different experiments, extreme cases analysis are analyzed. All DAX index components are used in the following experiments. In the whole experiment, the prediction results of experiments using Euclidean distance and correlation distance are relatively better than the results of experiments using City Block distance and cosine similarity. While the variances of log–returns of "future" part time series when using correlation distance and cosine similarity are higher than using Euclidean distance and City Block distances. Then we use the component itself only to construct sets of time frames in "history" part, the prediction results are slightly worse than the whole experiment but the prediction results are not completely random. This makes sense since when a better k nearest neighbors could be searched within a larger set of "history" time frames. The financial sector experiment uses the financial sector components as "history" and search k nearest neighbors of financial sector components only. The results of the financial sector experiment are quite similar to the results of individual experiments. The average predicted prices are similar but for the component with relatively high variance, the predicted prices are similar but the confidence interval is narrower. Comparing with the financial sector and individual experiment, it is obviously

that a smaller "history" set is used to achieve a similar result. Hence, the choice of "history" part is important in the kNN experiment. The kNN experiment of the financial sector is performed with selected components of FTSE100 from the financial sector such as banks, financial services etc. Compared to the DAX index financial sector experiment, more components are selected for FTSE100 index financial sector experiment. The results show that the for all distances and similarity measurement, the prediction results are relatively good since almost all real prices are within the confidence interval. The trend of predicted prices is closer to the trend of real prices. Therefore, selection of "history" is again playing a key role in this experiment. Hence, this approach of modeling and prediction using kNN works for ergodic dynamic systems. Comparison between the ARMA(1,1) model and kNN whole experiment shows that the prediction results are similar for the chosen components. The results also show that, when the ARMA(1,1) model has bad prediction results, the kNN experiment would have a bad prediction as well. Hence, some evidence could be concluded that this approach of using kNN is able to predict the random processes generated by using the ARMA model. Hence, we can conclude that using this kNN method, it shows that this approach works using data generated from the ARMA model. Since the nearest neighbor of a random process is simply a random process. Hence, it can conclude that this approach works for random process. This kNN historical Monte Carlo approach can be used to model or predict the daily log–returns. This algorithm is model free and the computation is relatively easy. In general, this method would predict ergodic time series if historical price time series is ergodic and this method predicts random process if historical prices are in a random process (i.e. white noise). However, optimization of parameters can be an interesting question to ask, i.e. values of k, the length of "present" and "future" part. The choice of "history" would be another interesting question to ask since kNN has limitation in storage of all distances between each point. Data generated from the Lorenz attractor with noise and random initial conditions are used in our kNN experiment with Euclidean distance. The Lorenz attractor is a deterministic system and the 'future' can be predicted. Our kNN experiment can find the k nearest neighbor predictions and for $k = 1$, the nearest neighbor case, we may still have some accurate predictions with small average RMSE. When $k \to 100$, the average RMSEs converge and the average standard deviation converge. This implies the convergence property of kNN as the density function converges when $k \to \infty$ and $n \to \infty$. From average standard deviation results, a sharp peak exists at the end. It may suggest that there is a limitation on how long can we predict for a fixed 'present' (i.e. if we

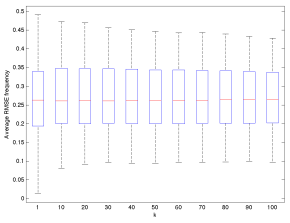
have 'present' length of 20, the prediction power of 'future' length of more than 20 is weak).



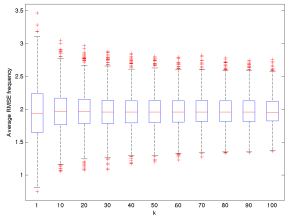
(A) $X(t)$, $\alpha_\epsilon = 0$



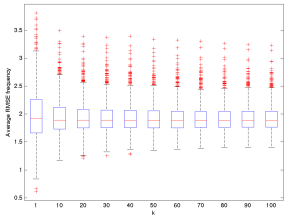
(B) $Y(t)$, $\alpha_\epsilon = 0$



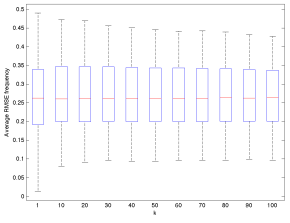
(C) $Z(t)$, $\alpha_\epsilon = 0$



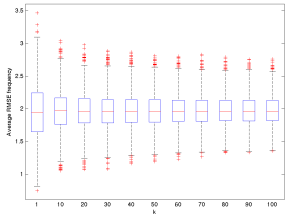
(D) $X(t)$, $\alpha_\epsilon = 0.05$



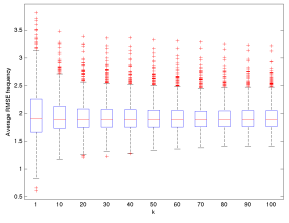
(E) $Y(t)$, $\alpha_\epsilon = 0.05$



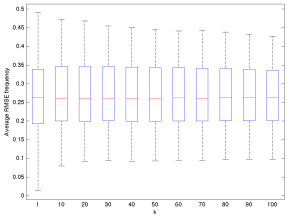
(F) $Z(t)$, $\alpha_\epsilon = 0.05$



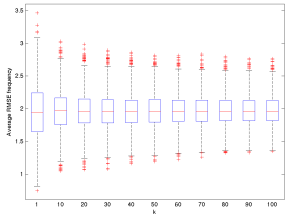
(G) $X(t)$, $\alpha_\epsilon = 0.1$



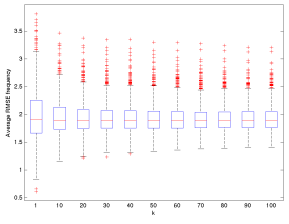
(H) $Y(t)$, $\alpha_\epsilon = 0.1$



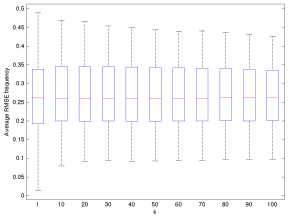
(I) $Z(t)$, $\alpha_\epsilon = 0.1$



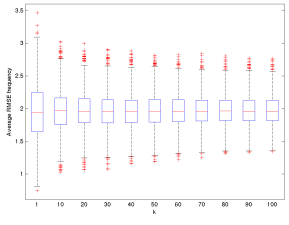
(J) $X(t)$, $\alpha_\epsilon = 0.15$



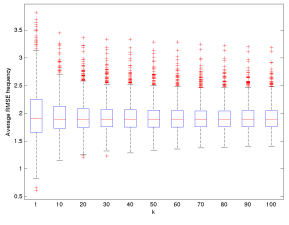
(K) $Y(t)$, $\alpha_\epsilon = 0.15$



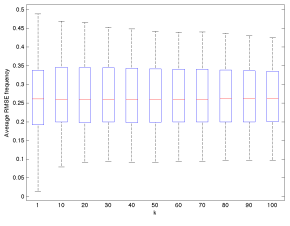
(L) $Z(t)$, $\alpha_\epsilon = 0.15$



(M) $X(t)$, $\alpha_\epsilon = 0.2$

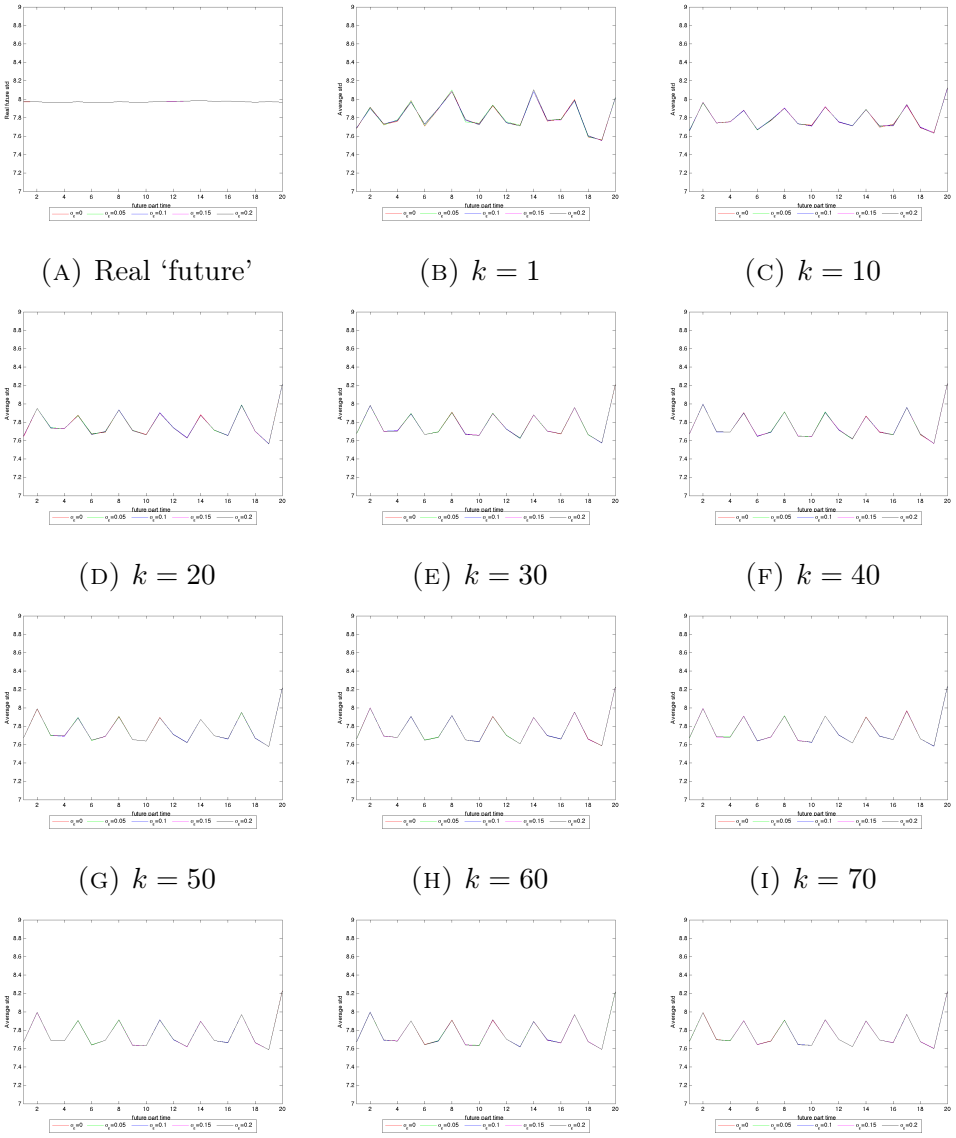


(N) $Y(t)$, $\alpha_\epsilon = 0.2$



(O) $Z(t)$, $\alpha_\epsilon = 0.2$

FIGURE 3.51: Boxplots of average RMSE for all k for different solution and different α_ϵ.



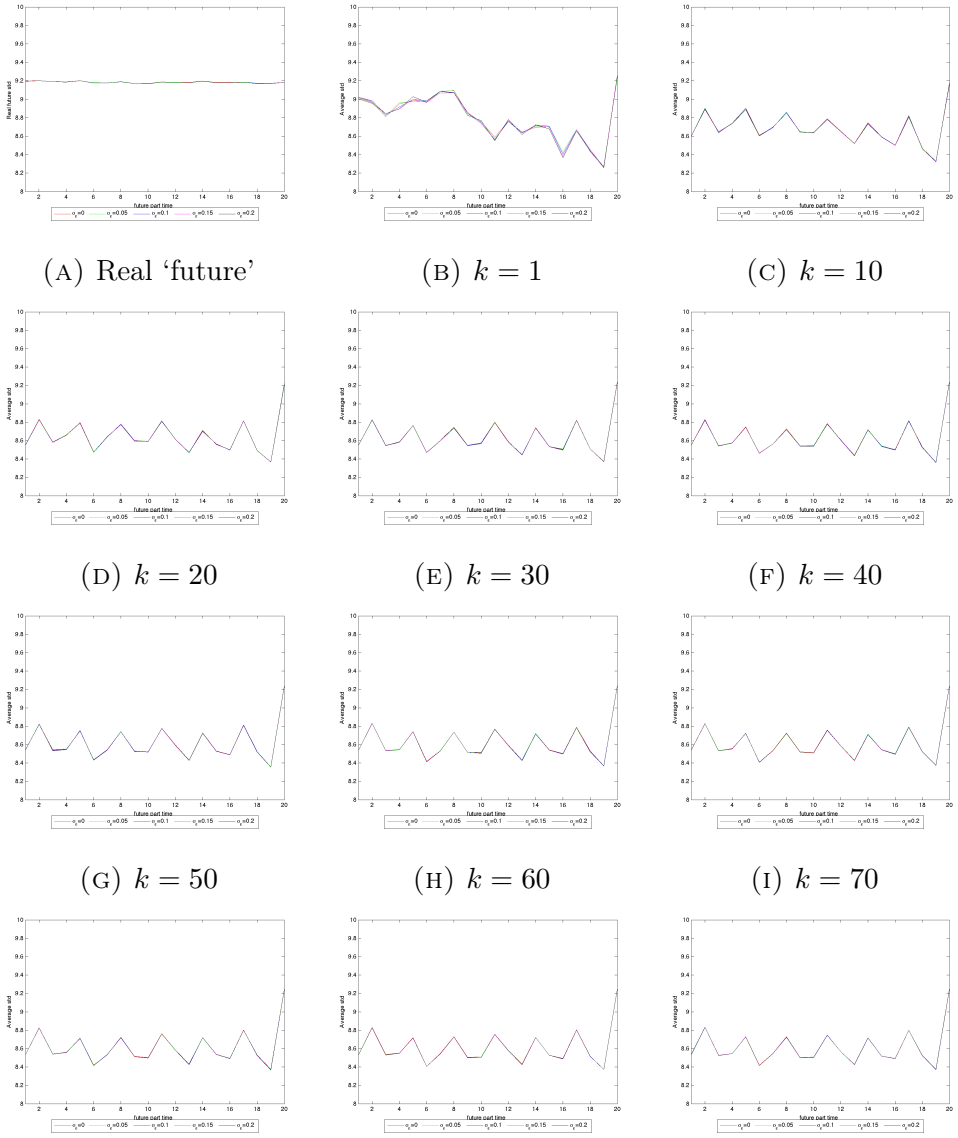
(A) Real 'future' (B) $k = 1$ (C) $k = 10$

(D) $k = 20$ (E) $k = 30$ (F) $k = 40$

(G) $k = 50$ (H) $k = 60$ (I) $k = 70$

(J) $k = 80$ (K) $k = 90$ (L) $k = 100$

FIGURE 3.52: Standard deviation of all 'windows' real 'future' data and average of k prediction SD of all 'windows' against 'future' time t for solution $X(t)$.



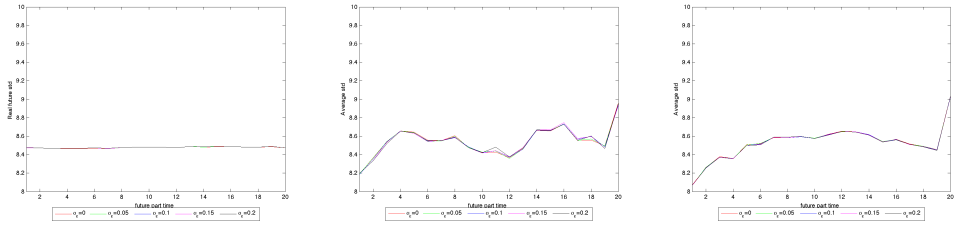
(A) Real 'future' (B) $k = 1$ (C) $k = 10$

(D) $k = 20$ (E) $k = 30$ (F) $k = 40$

(G) $k = 50$ (H) $k = 60$ (I) $k = 70$

(J) $k = 80$ (K) $k = 90$ (L) $k = 100$

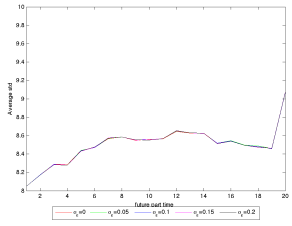
FIGURE 3.53: Standard deviation of all 'windows' real 'future' data and average of k prediction SD of all 'windows' against 'future' time t for solution $Y(t)$.



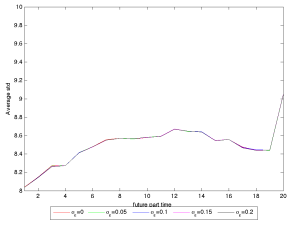
(A) Real 'future'

(B) $k = 1$

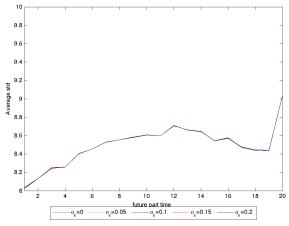
(C) $k = 10$



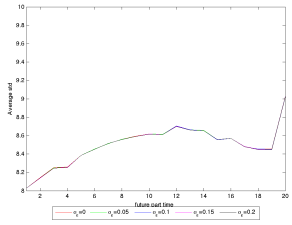
(D) $k = 20$



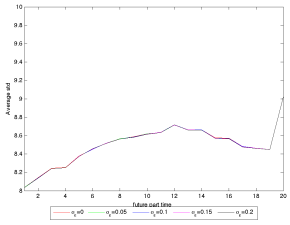
(E) $k = 30$



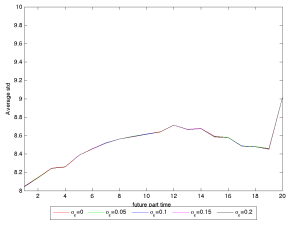
(F) $k = 40$



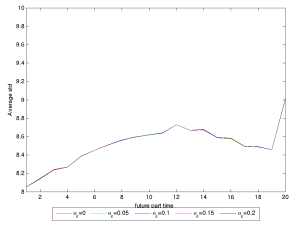
(G) $k = 50$



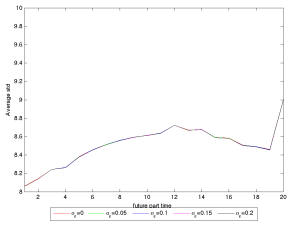
(H) $k = 60$



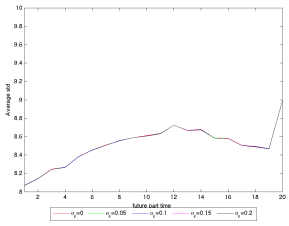
(I) $k = 70$



(J) $k = 80$



(K) $k = 90$



(L) $k = 100$

FIGURE 3.54: Standard deviation of all 'windows' real 'future' data and average of k prediction SD of all 'windows' against 'future' time t for solution $Z(t)$.

Chapter 4

Conclusion and future direction

In this thesis, applications of kNN rule in stock and share time series are studied. We aim to develop approaches using kNN rule to analyze the predictability of this time series and approaches using kNN to model and predict stock and share time series. We believe that there are 2 assumptions of the markets. The first assumption believes that 'history repeats itself' which means historical prices have information on future prices. Hence, the market is profitable by applying indicators and methods from technical analysis. The second assumption the market is efficient and it is not possible to gain profits by using historical prices. This is the idea of EMH and this contradicts the first assumption that with this assumption, the prices are random. In Chapter 2, the components of four market indices are labeled with 'winner' or 'loser' based on the ordering strategy we created. We consider the prices of initial 3 months and last 3 months and compute the average prices and average price ratios. Then we reorder the components with descending order of average price ratios and label the first 1/3 to be 'winner' and last 'loser'. From the result of LOOCV error analysis, the LOOCV errors for components of HANGSENG index and DAX index are lower. Hence, it concludes that components of HANGSENG and DAX indices have positive predictability. And it further implies that technical analysis could be applied and have a higher possibility to gain profit. The LOOCV errors for FTSE and NASDAQ index are much higher. This could mean that the market prices are independent in terms of long term success and therefore it further implies that they are efficient markets. The 2–D and 3–D principal manifold graphs generated from VidaExpert show that for HANGSENG and DAX index, the 'winner' and 'loser' components are well separated and clusters are formed. For NASDAQ and FTSE index, there is no sign of separation on the manifold plots. Therefore, it can be concluded that for a

young market as HANGSENG and DAX index, there are positive predictabilities and technical analysis could be applied to obtain a high possibility of profit. For mature markets as NASDAQ and FTSE index, the markets are efficient and there is no predictability in terms of long–term success. Since clusters of 'winner' and 'loser' are shown in the principal manifold graphs, an interesting future research direction based on this would be cluster analysis on components that are closed together for positive predictability time series. There may be a link between the structure of cluster with predictability and it is interesting to study on this question. The distance ratio between centroid of 'winner' and 'loser' can be studied in relation with the length of time interval chosen in the beginning to optimize our results. In Chapter 3, we aimed to develop a kNN–based universal indicator for both ergodic dynamic system (under the assumption of technical analysis) and random process (under the assumption of EMH). Daily log–returns are computed from closing prices. The log–returns time series is split into 3 parts, "history" part, "present" part and "future" part. The training set is constructed from "history" part from a set of shifted time frames of "history" part time series. k nearest neighbors of "present" part is searched from this training set and these are considered as predicted "future" part log–return factors. The log–return factors are then transformed back to closing prices to construct the predicted "future" part time series. The data are closing prices of DAX index components and FTSE100 index components from financial sectors. The different experiment uses a slightly different training set to search for nearest neighbors. For a preset $k = 60$, "present" part length is 60 and "future" part length is 40, the kNN experiment with training set from whole experiment have better prediction results. For FTSE100 components from financial sectors, the prediction results are ok. Four measurements of distances are applied for all experiments. It shows that the results of Euclidean distance and City Block distance are similar. The results of experiments using correlation distance and cosine similarity are similar. In general, the Euclidean distance and City Block distance results have better confidence interval while the correlation distance and cosine similarity gives a better average prediction prices. At this stage, by analyzing 3 extreme cases, our approach work well for ergodic dynamic system time series. Hence, we conclude that the historical prices contain information on future prices and this approaches works in a predictable market, or 'world of technical analysis'. In the end, the ARMA$(1, 1)$ model is applied and the prediction results are compared with the kNN experiment. From the graphs, the predicted closing prices and confidence intervals are quite similar. Hence, this would indicate that our kNN approach works for stationary time series and it

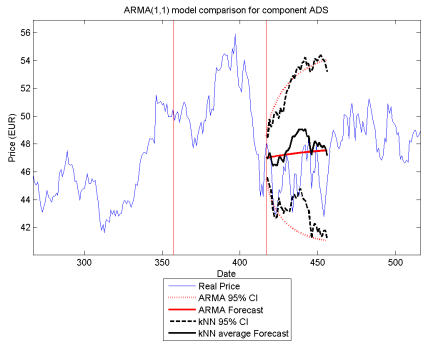
indicates that this approach also works for random process or 'world of EMH'.

Data generated from the Lorenz attractor with Gaussian noise are used for the kNN experiment. The test set and training set are chosen and sliding window strategy is applied on test set and training set. k nearest neighbors are computed and the next 'window' is used as the prediction for 'future'. For each solution, the k nearest neighbors are computed for each 'window' and they are chosen from the 'history' part of each solution. Average RMSE is used to measure the error between the predicted values and real values. From the results, it shows that different level of noise does not make many differences in average RMSE and when $k \to 100$, the average RMSE values converge. When $k = 1$, average RMSE can be very small. The prediction for solution $Z(t)$ is best overall. Average standard deviations are applied to measure the spread of predictions for all 'windows'. There is a peak in the end for all three cases and this would show that there is a limit of prediction power. This would imply that the 'future' time 'window' lengths should not be larger than the 'window' length. When $k \to \infty$, the average standard deviations converge. This can be explained as the convergence property of kNN.

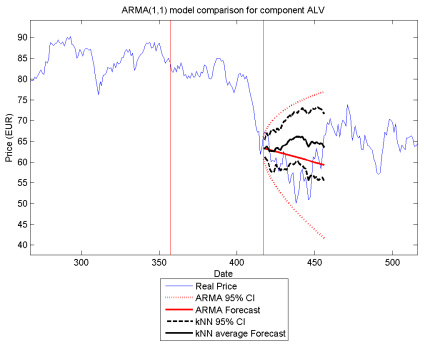
A good extension of the question could be the optimization of parameters for kNN approach. Five parameters are used in the KNN experiment (i.e. k, the measure of distance, choice of "history" part, "present" part length and "future" part length). The methods to developed a best k can be studied since kNN has a lack of storage of distance. Since for longer lengths of "future" part, in our experiment, the results are not getting better and the variances of log–return factors seem to converge. Another interesting extension could be using modified nearest neighbor techniques to fight against the limitation of kNN and apply weights on kNN to solve the border issue.

Appendix A

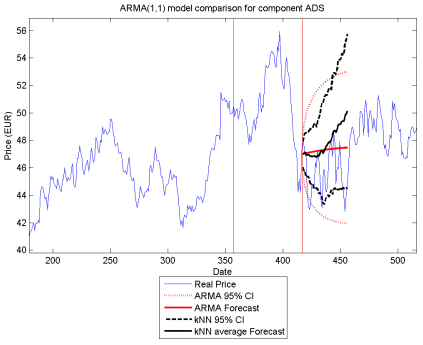
Graphs of comparison between the kNN approach and the ARMA(1,1) model (Euclidean Distance)



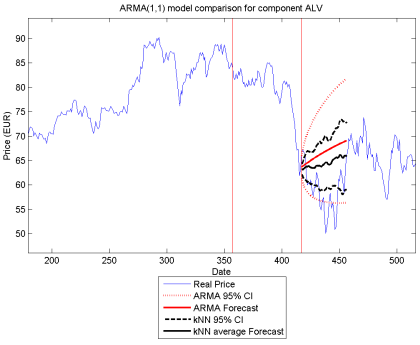
(A) 25% "history" ADS



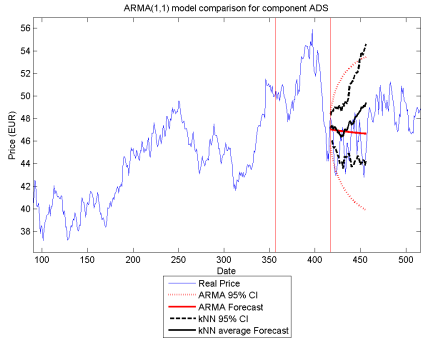
(B) 25% "history" ALV



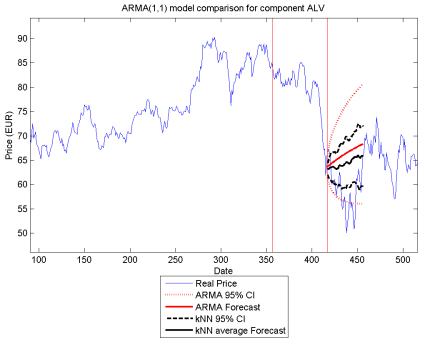
(C) 50% "history" ADS



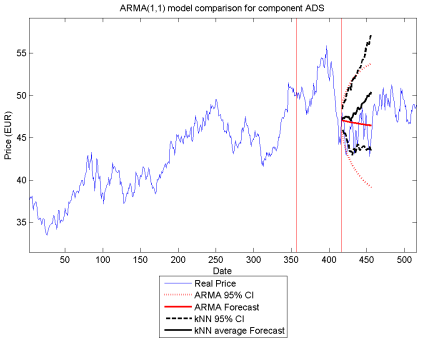
(D) 50% "history" ALV



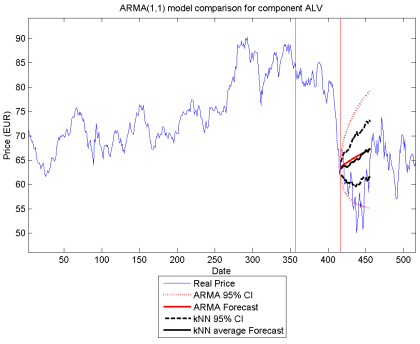
(E) 75% "history" ADS



(F) 75% "history" ALV

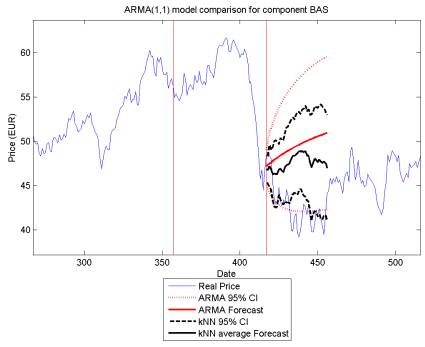


(G) 100% "history" ADS

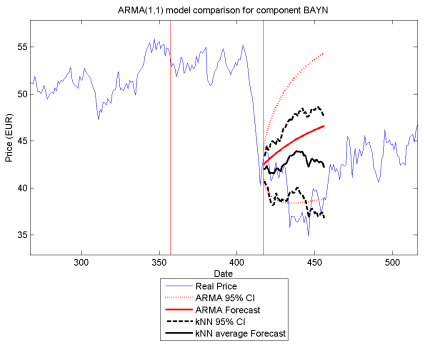


(H) 100% "history" ALV

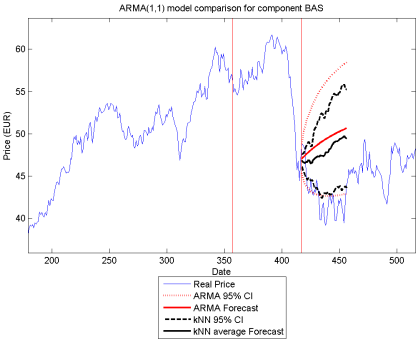
FIGURE A.1: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component ADS (all LEFT figures) and ALV (all RIGHT figures)



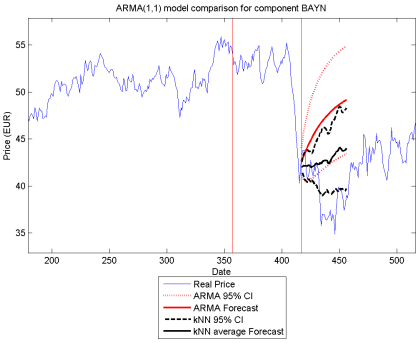
(A) 25% "history" BAS



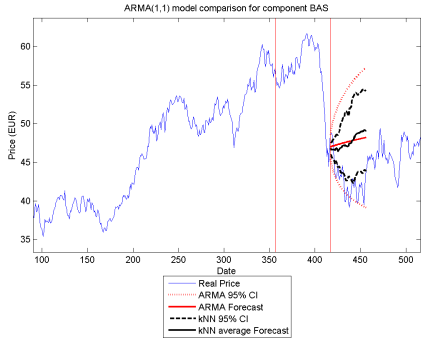
(B) 25% "history" BAYN



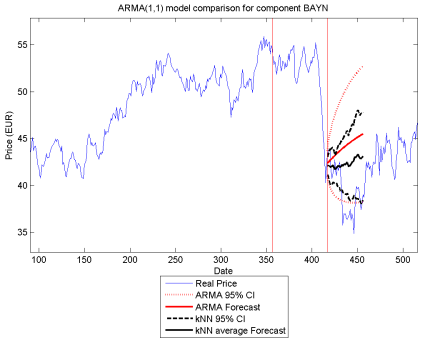
(C) 50% "history" BAS



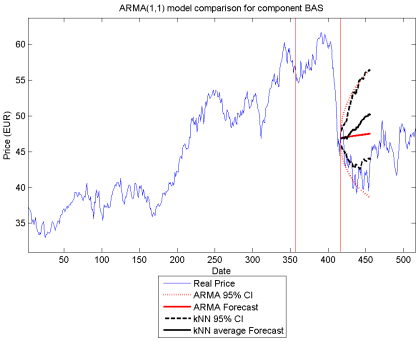
(D) 50% "history" BAYN



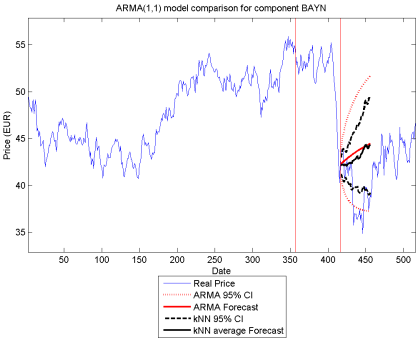
(E) 75% "history" BAS



(F) 75% "history" BAYN

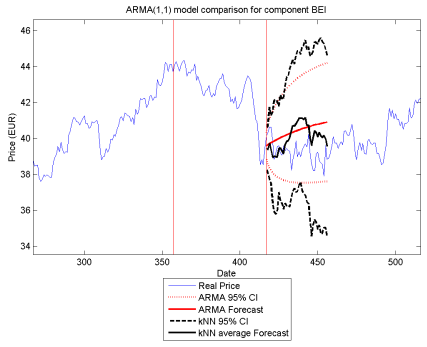


(G) 100% "history" BAS

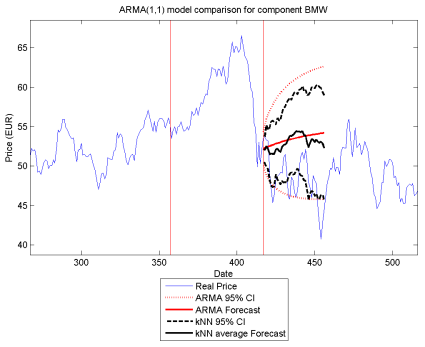


(H) 100% "history" BAYN

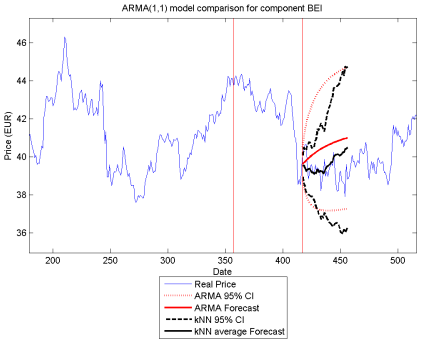
FIGURE A.2: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component BAS (all LEFT figures) and BAYN (all RIGHT figures)



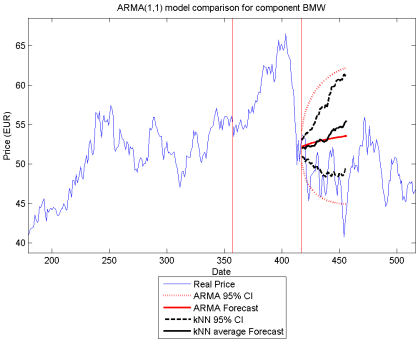
(A) 25% "history" BEI



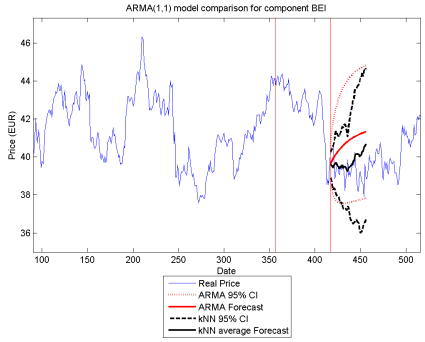
(B) 25% "history" BMW



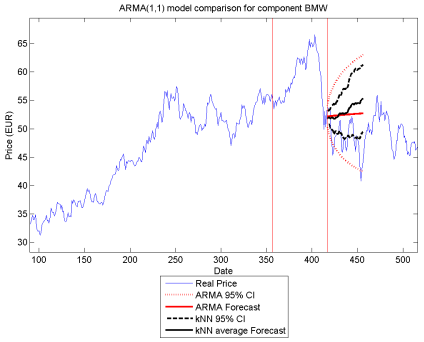
(C) 50% "history" BEI



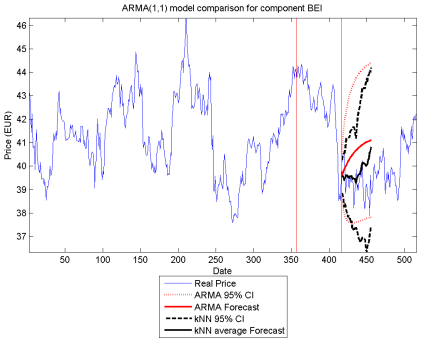
(D) 50% "history" BMW



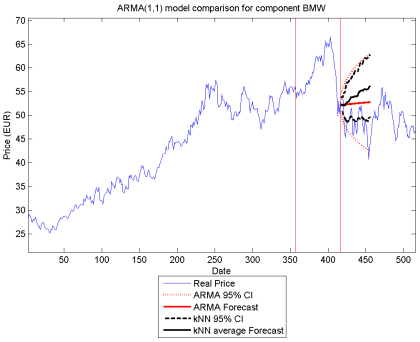
(E) 75% "history" BEI



(F) 75% "history" BMW

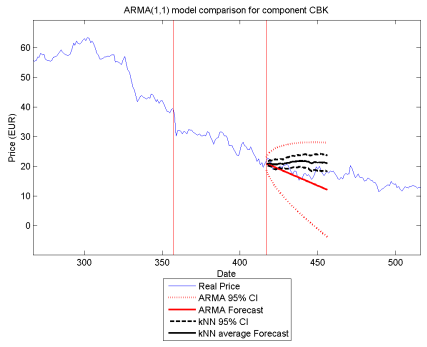


(G) 100% "history" BEI

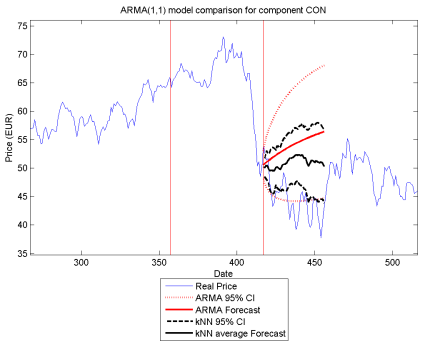


(H) 100% "history" BMW

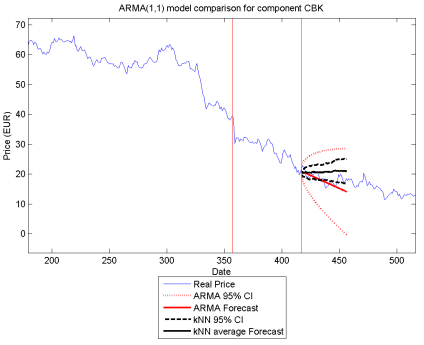
FIGURE A.3: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component BEI (all LEFT figures) and BMW (all RIGHT figures)



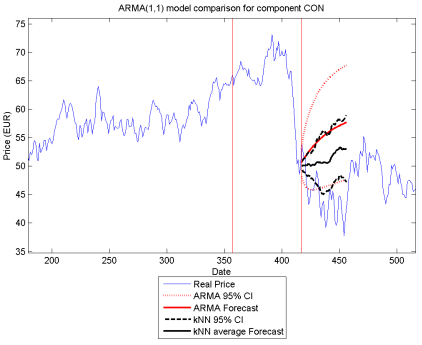
(A) 25% "history" CBK



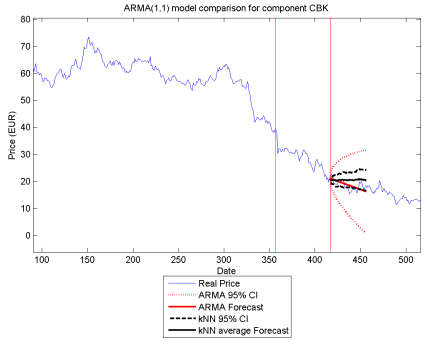
(B) 25% "history" CON



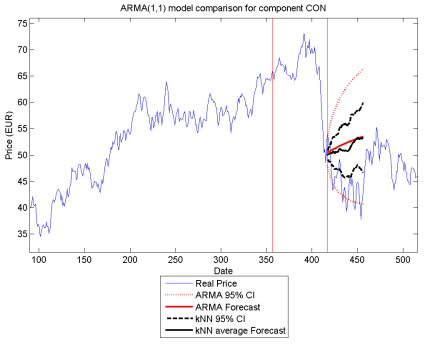
(C) 50% "history" CBK



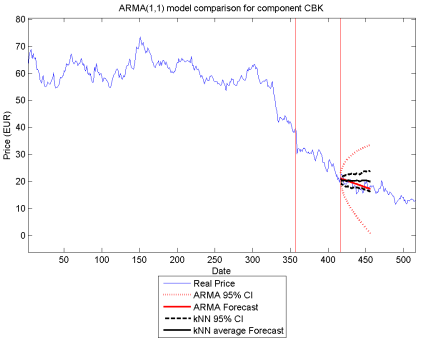
(D) 50% "history" CON



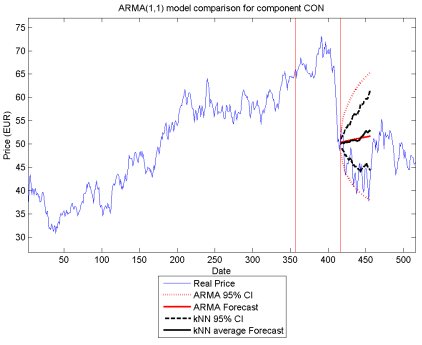
(E) 75% "history" CBK



(F) 75% "history" CON

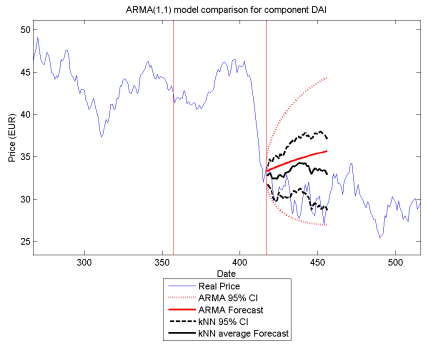


(G) 100% "history" CBK

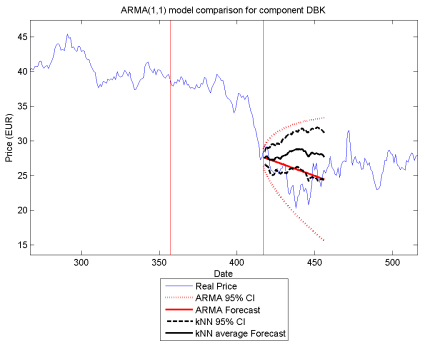


(H) 100% "history" CON

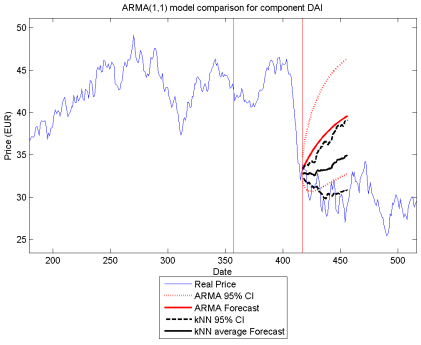
FIGURE A.4: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component CBK (all LEFT figures) and CON (all RIGHT figures)



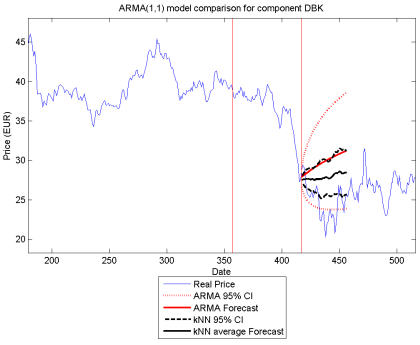
(A) 25% "history" DAI



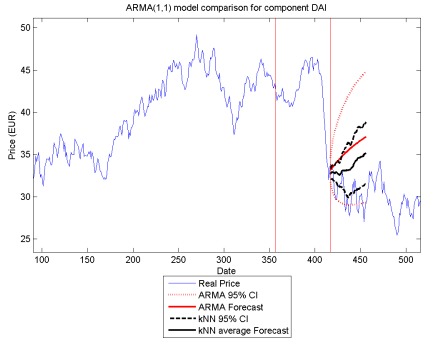
(B) 25% "history" DBK



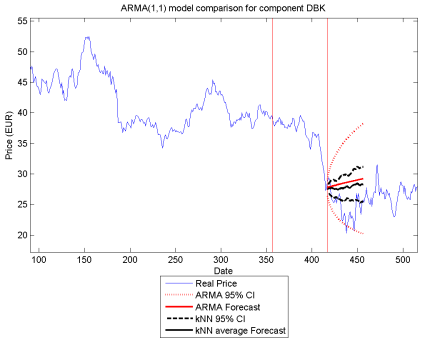
(C) 50% "history" DAI



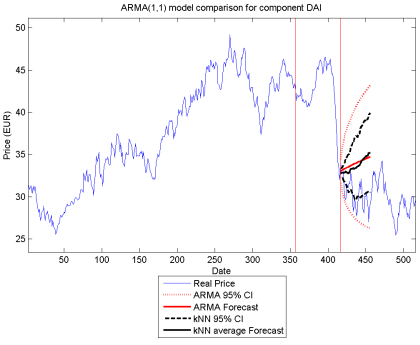
(D) 50% "history" DBK



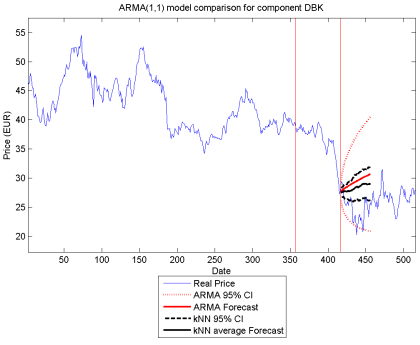
(E) 75% "history" DAI



(F) 75% "history" DBK

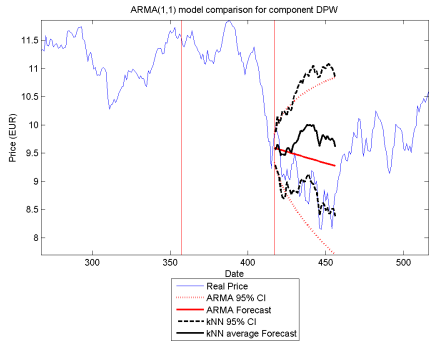


(G) 100% "history" DAI

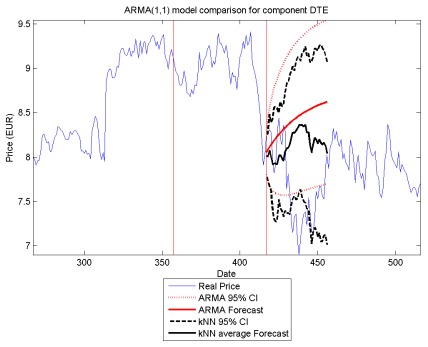


(H) 100% "history" DBK

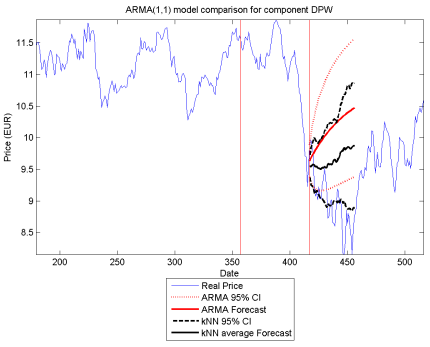
FIGURE A.5: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component DAI (all LEFT figures) and DBK (all RIGHT figures)



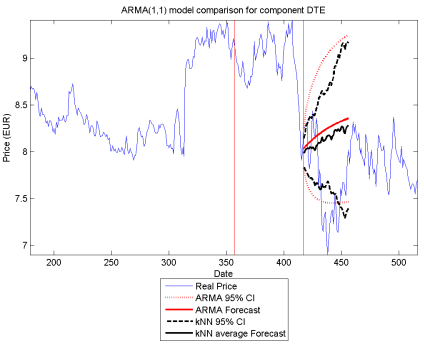
(A) 25% "history" DPW



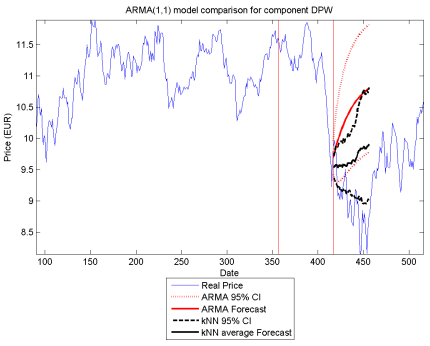
(B) 25% "history" DTE



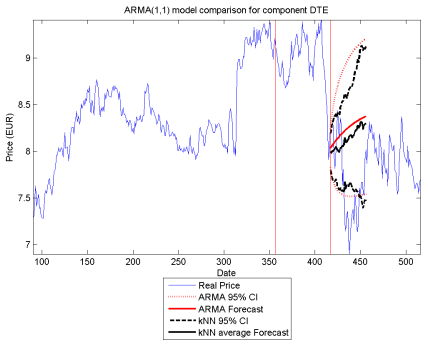
(C) 50% "history" DPW



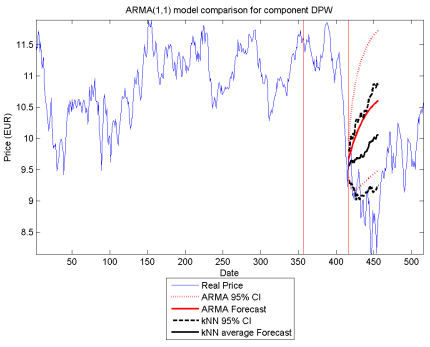
(D) 50% "history" DTE



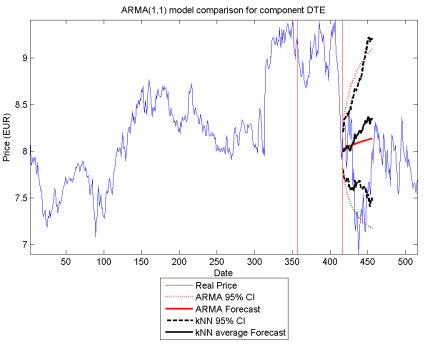
(E) 75% "history" DPW



(F) 75% "history" DTE

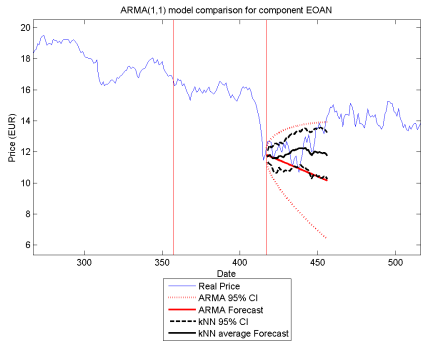


(G) 100% "history" DPW

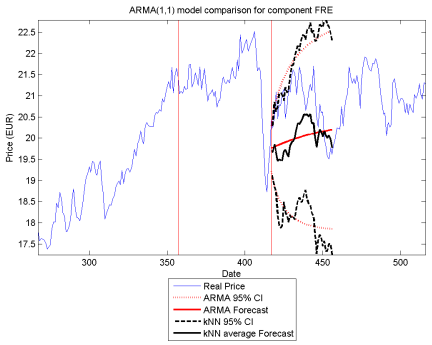


(H) 100% "history" DTE

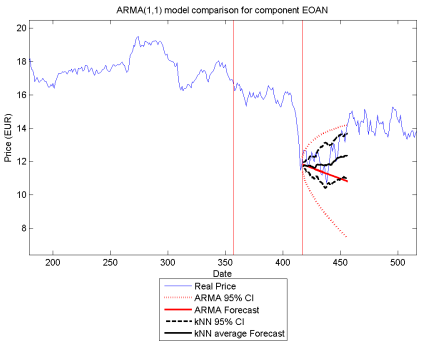
FIGURE A.6: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component DPW (all LEFT figures) and DTE (all RIGHT figures)



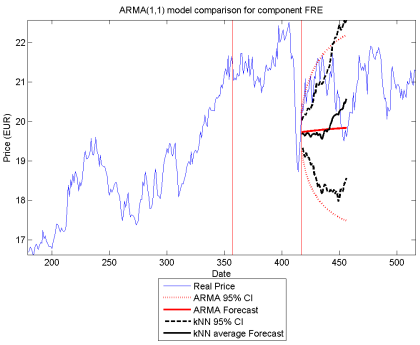
(A) 25% "history" EOAN



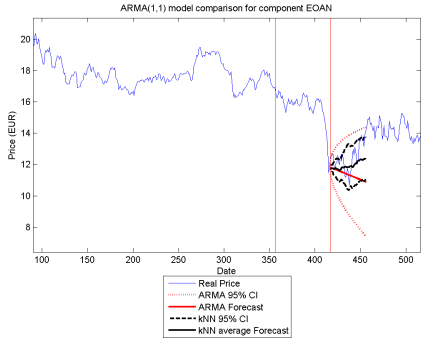
(B) 25% "history" FRE



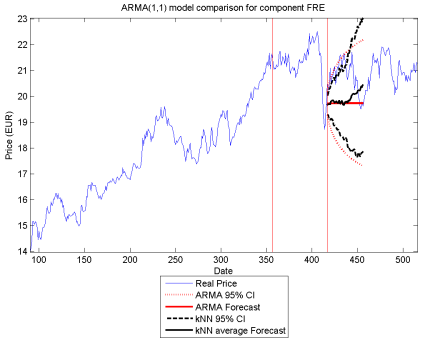
(C) 50% "history" EOAN



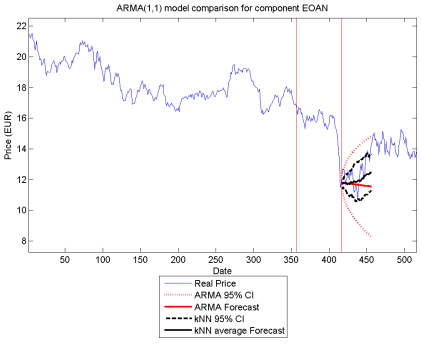
(D) 50% "history" FRE



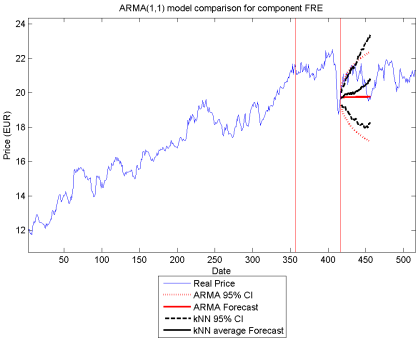
(E) 75% "history" EOAN



(F) 75% "history" FRE

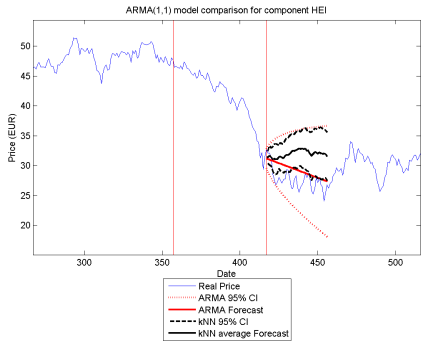


(G) 100% "history" EOAN

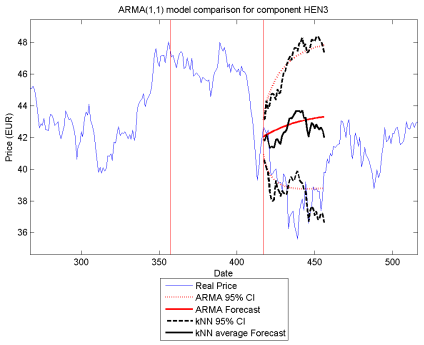


(H) 100% "history" FRE

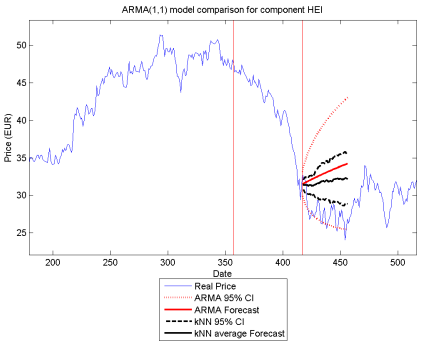
FIGURE A.7: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component EOAN (all LEFT figures) and FRE (all RIGHT figures)



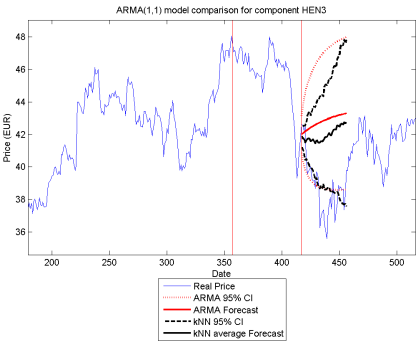
(A) 25% "history" HEI



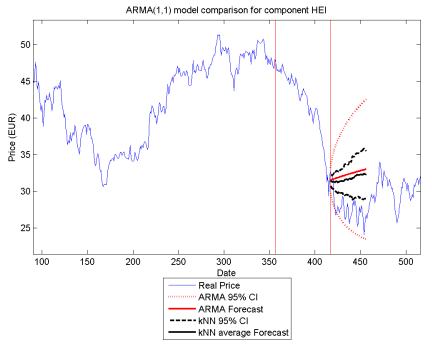
(B) 25% "history" HEN3



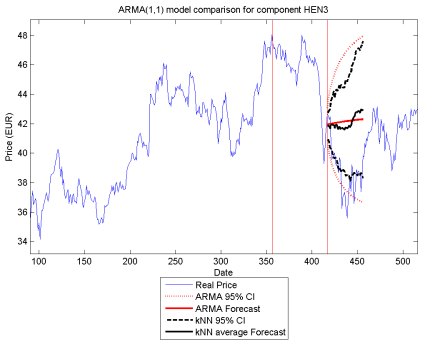
(C) 50% "history" HEI



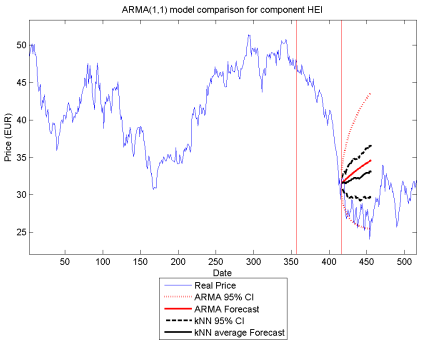
(D) 50% "history" HEN3



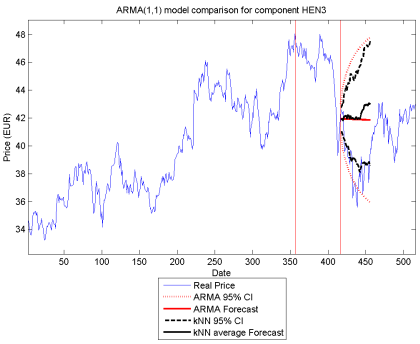
(E) 75% "history" HEI



(F) 75% "history" HEN3

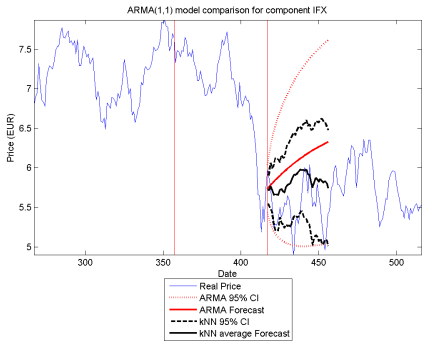


(G) 100% "history" HEI

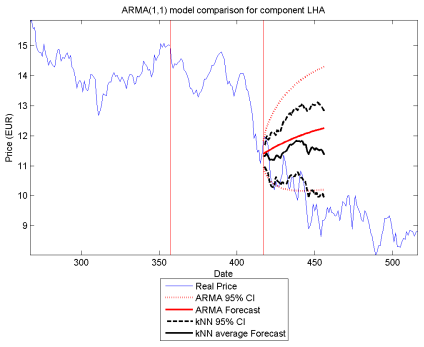


(H) 100% "history" HEN3

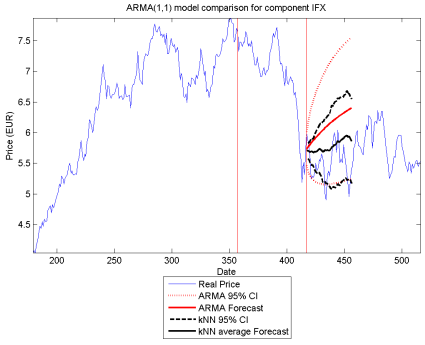
FIGURE A.8: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component HEI (all LEFT figures) and HEN3 (all RIGHT figures)



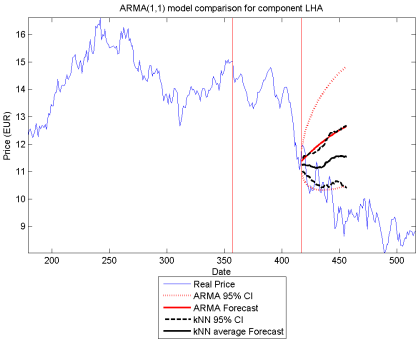
(A) 25% "history" IFX



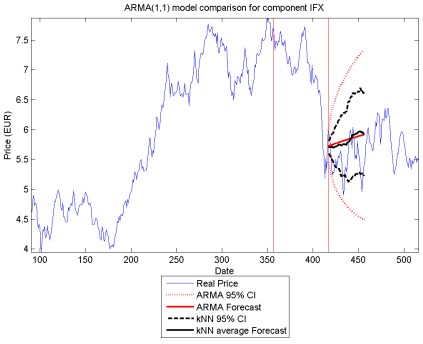
(B) 25% "history" LHA



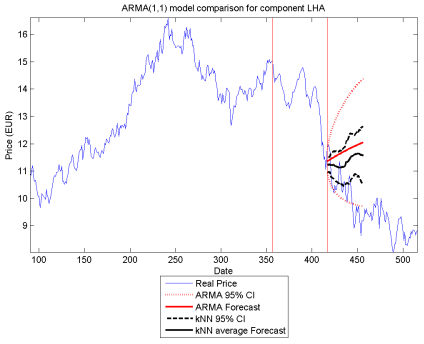
(C) 50% "history" IFX



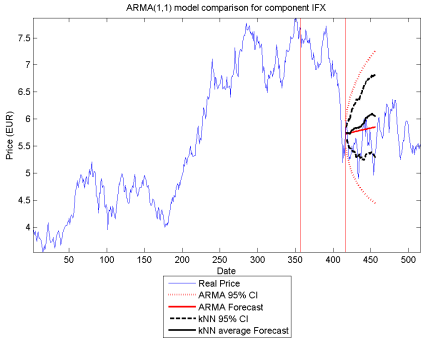
(D) 50% "history" LHA



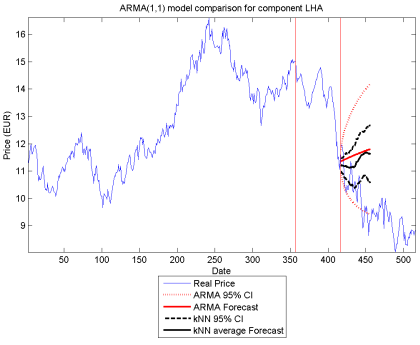
(E) 75% "history" IFX



(F) 75% "history" LHA

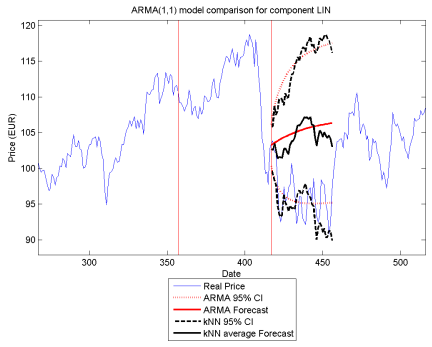


(G) 100% "history" IFX

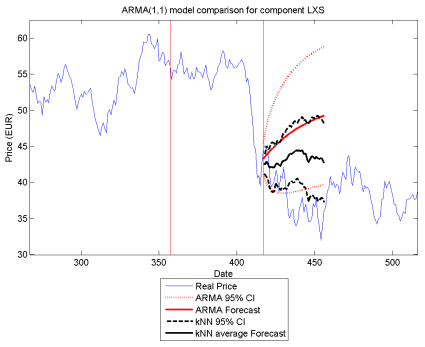


(H) 100% "history" LHA

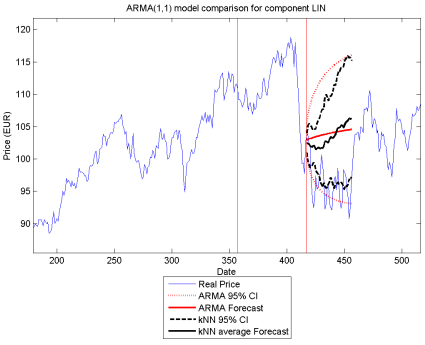
FIGURE A.9: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component IFX (all LEFT figures) and LHA (all RIGHT figures)



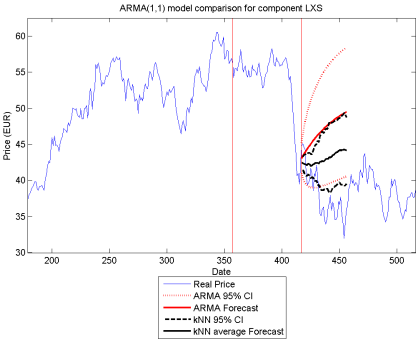
(A) 25% "history" LIN



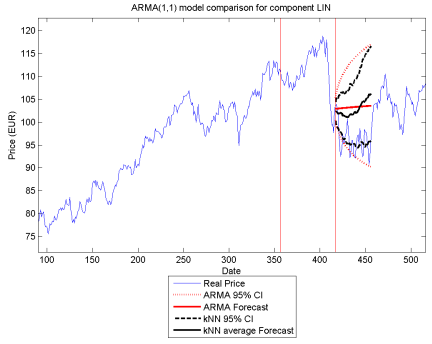
(B) 25% "history" LXS



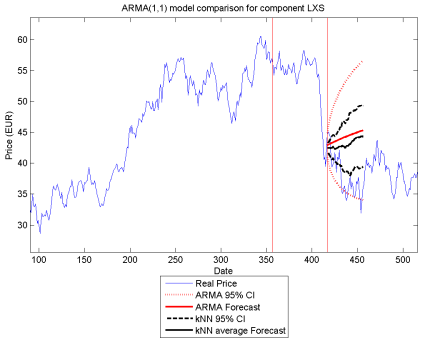
(C) 50% "history" LIN



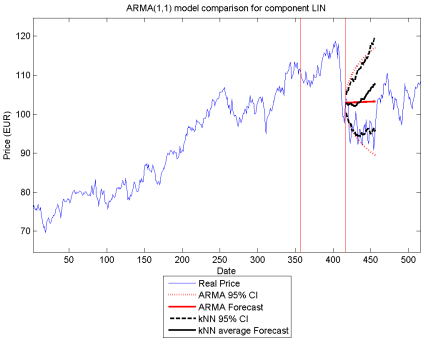
(D) 50% "history" LXS



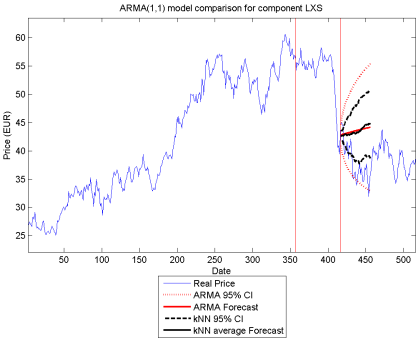
(E) 75% "history" LIN



(F) 75% "history" LXS

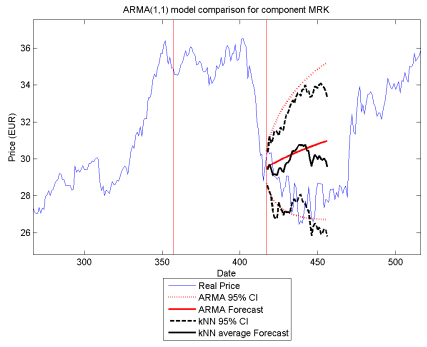


(G) 100% "history" LIN

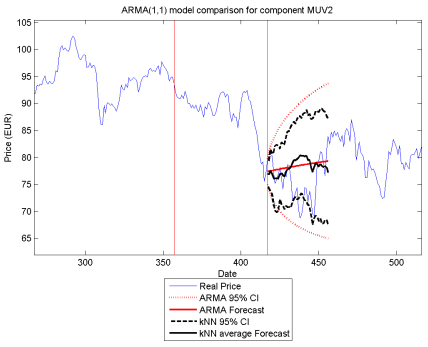


(H) 100% "history" LXS

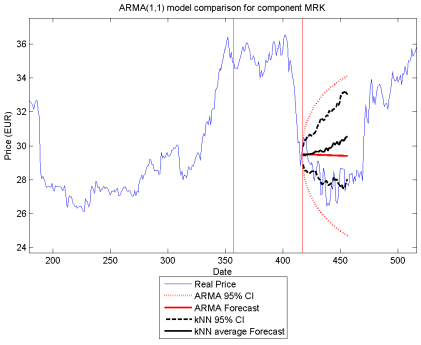
FIGURE A.10: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component LIN (all LEFT figures) and LXS (all RIGHT figures)



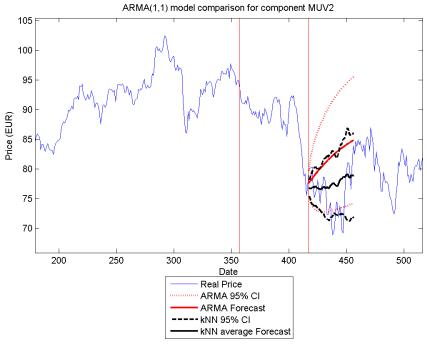
(A) 25% "history" MRK



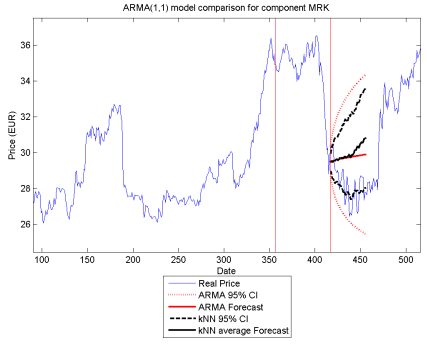
(B) 25% "history" MUV2



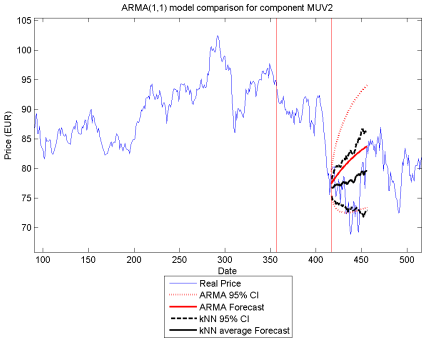
(C) 50% "history" MRK



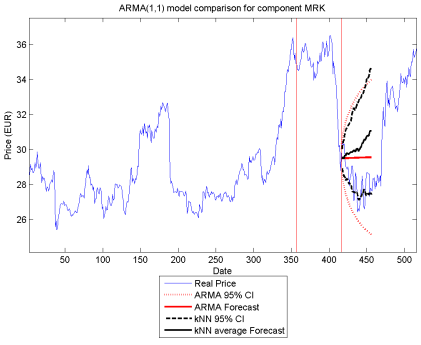
(D) 50% "history" MUV2



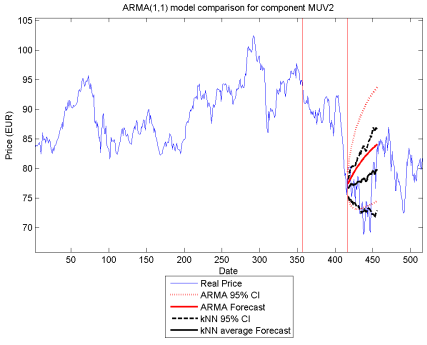
(E) 75% "history" MRK



(F) 75% "history" MUV2

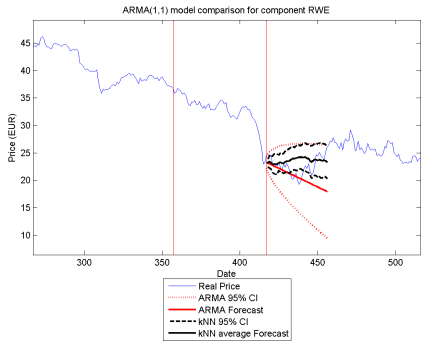


(G) 100% "history" MRK

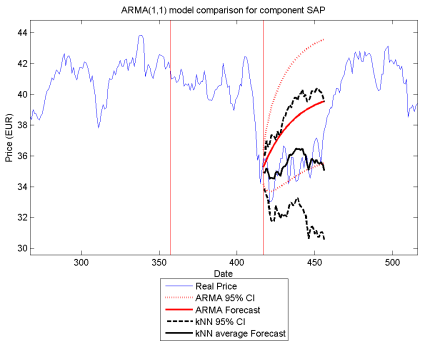


(H) 100% "history" MUV2

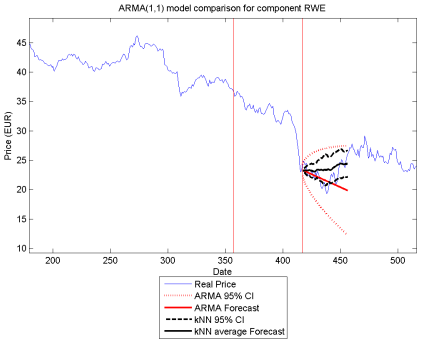
FIGURE A.11: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component MRK (all LEFT figures) and MUV2 (all RIGHT figures)



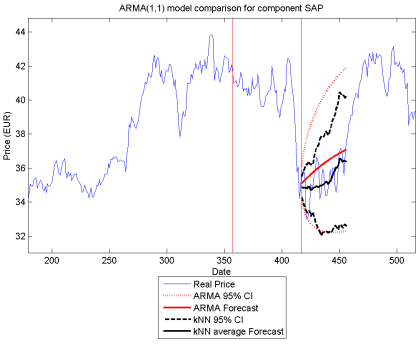
(A) 25% "history" RWE



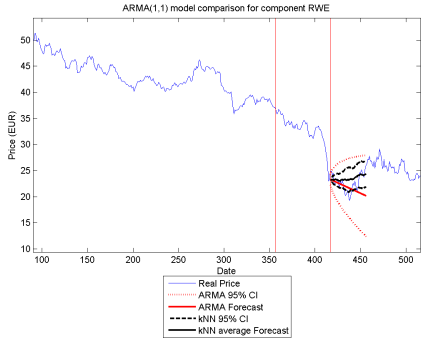
(B) 25% "history" SAP



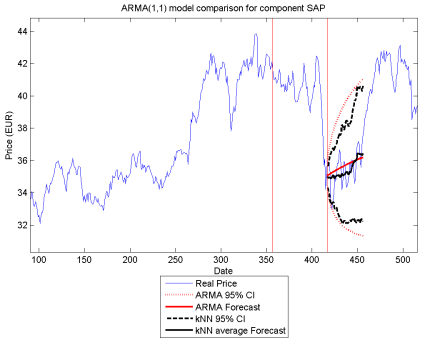
(C) 50% "history" RWE



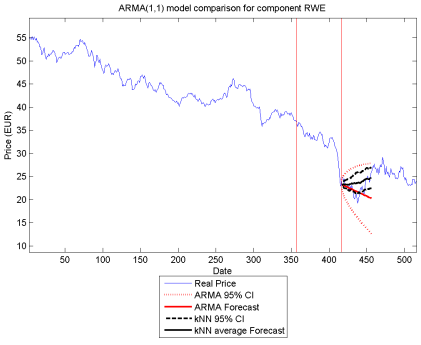
(D) 50% "history" SAP



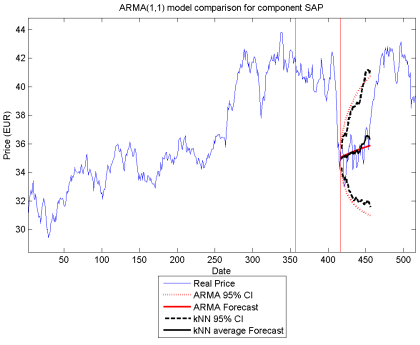
(E) 75% "history" RWE



(F) 75% "history" SAP

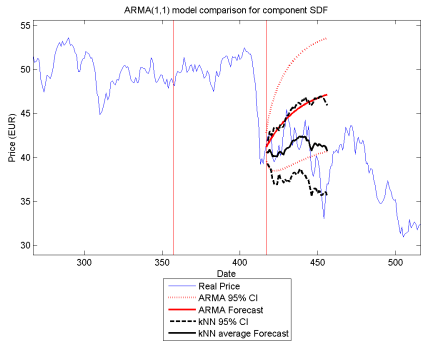


(G) 100% "history" RWE

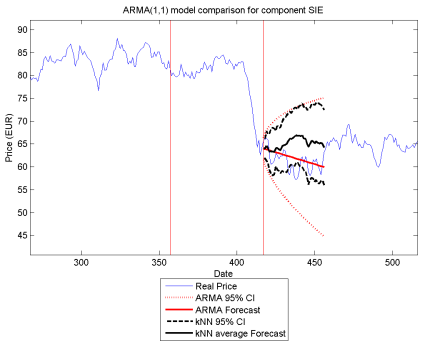


(H) 100% "history" SAP

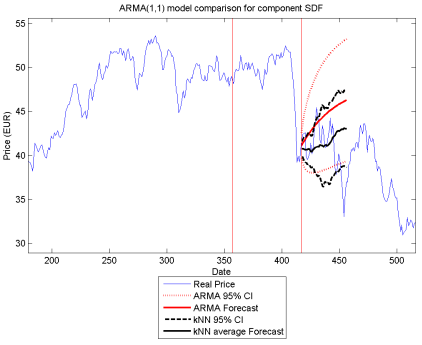
FIGURE A.12: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component RWE (all LEFT figures) and SAP (all RIGHT figures)



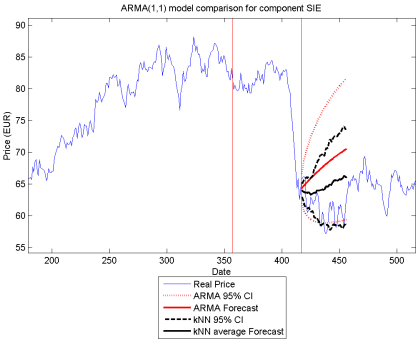
(A) 25% "history" SDF



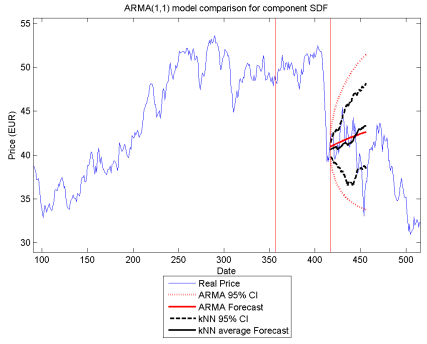
(B) 25% "history" SIE



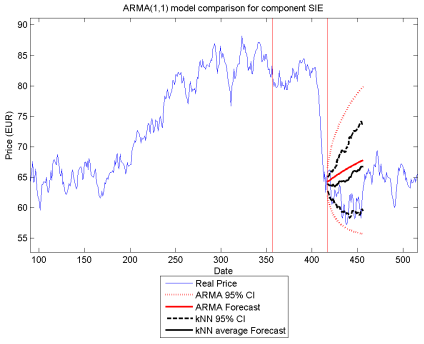
(C) 50% "history" SDF



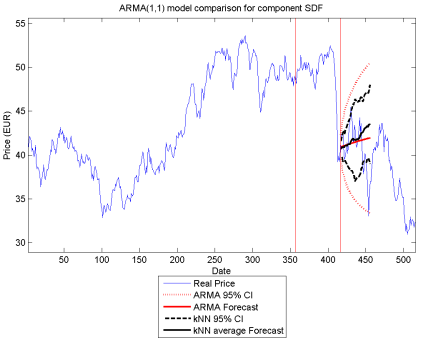
(D) 50% "history" SIE



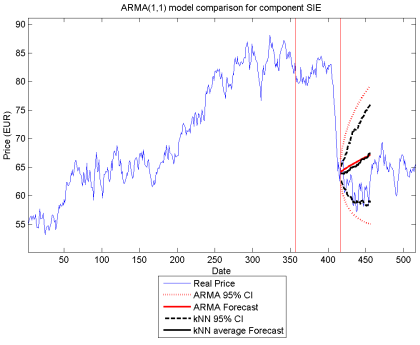
(E) 75% "history" SDF



(F) 75% "history" SIE

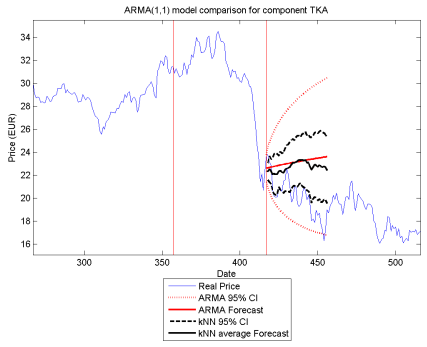


(G) 100% "history" SDF

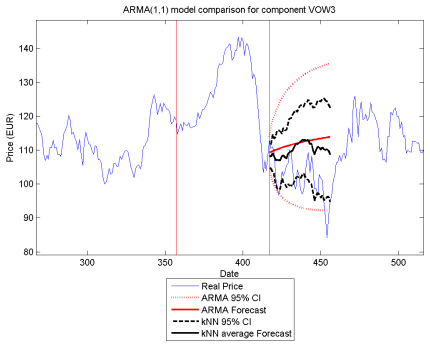


(H) 100% "history" SIE

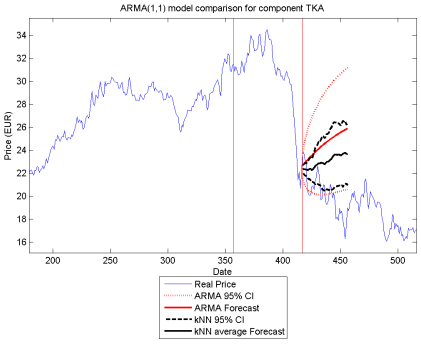
FIGURE A.13: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component SDF (all LEFT figures) and SIE (all RIGHT figures)



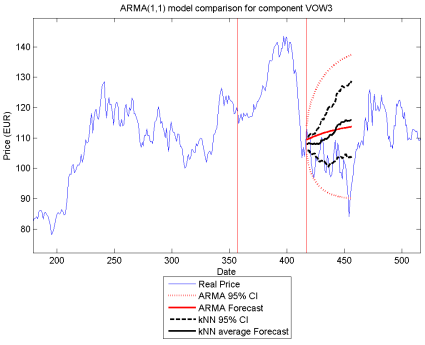
(A) 25% "history" TKA



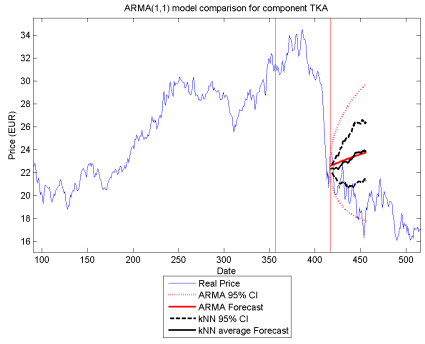
(B) 25% "history" VOW3



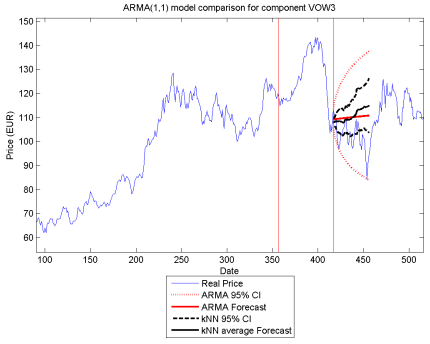
(C) 50% "history" TKA



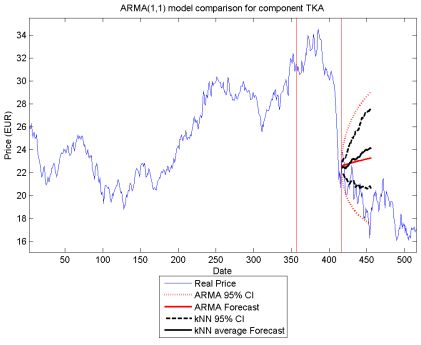
(D) 50% "history" VOW3



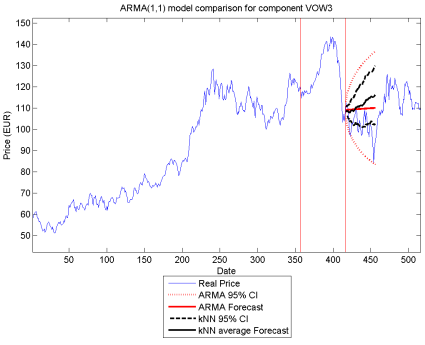
(E) 75% "history" TKA



(F) 75% "history" VOW3



(G) 100% "history" TKA

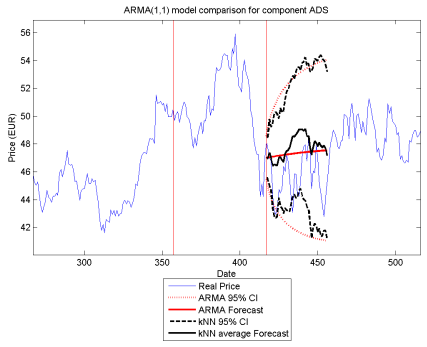


(H) 100% "history" VOW3

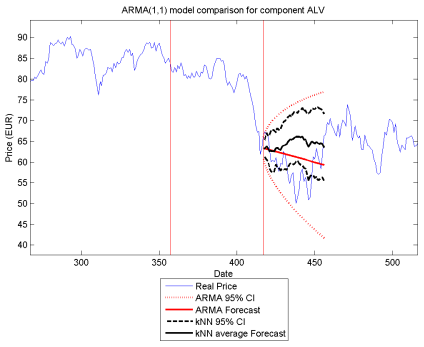
FIGURE A.14: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component TKA (all LEFT figures) and VOW3 (all RIGHT figures)

Appendix B

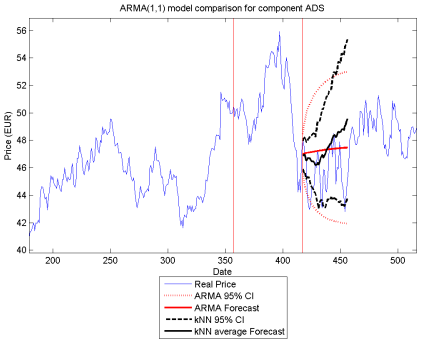
Graphs of comparison between the kNN approach and the ARMA(1,1) model (City Block Distance)



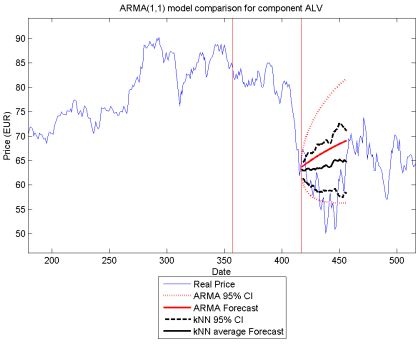
(A) 25% "history" ADS



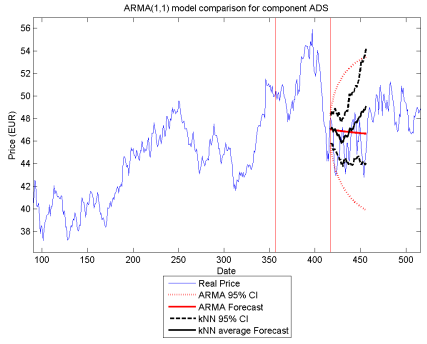
(B) 25% "history" ALV



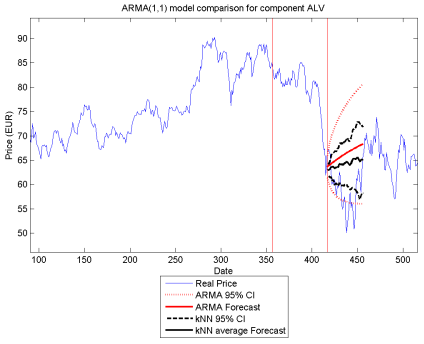
(C) 50% "history" ADS



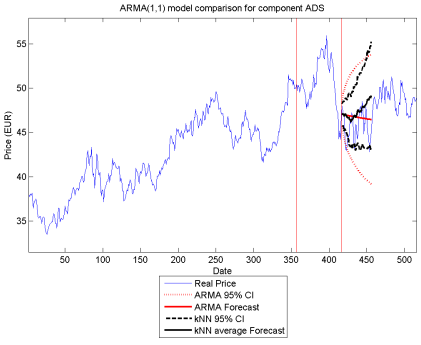
(D) 50% "history" ALV



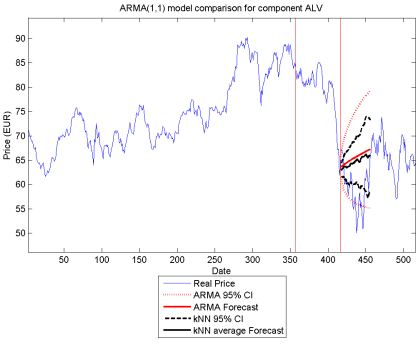
(E) 75% "history" ADS



(F) 75% "history" ALV

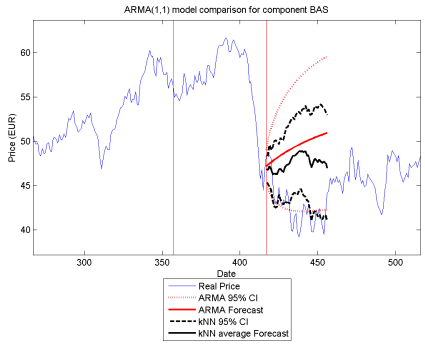


(G) 100% "history" ADS

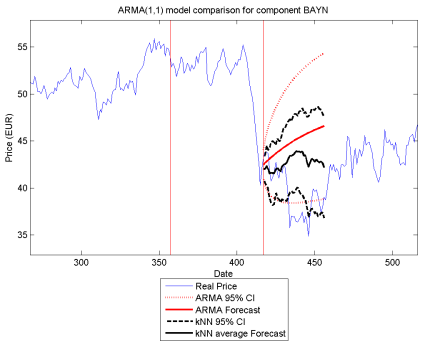


(H) 100% "history" ALV

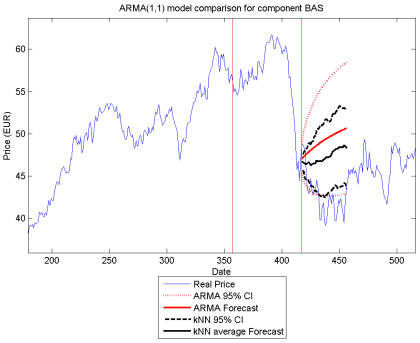
FIGURE B.1: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component ADS (all LEFT figures) and ALV (all RIGHT figures)



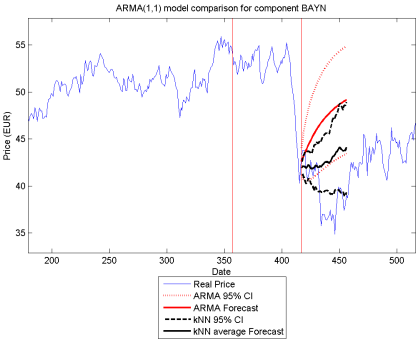
(A) 25% "history" BAS



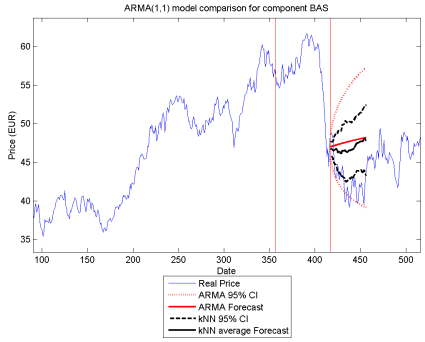
(B) 25% "history" BAYN



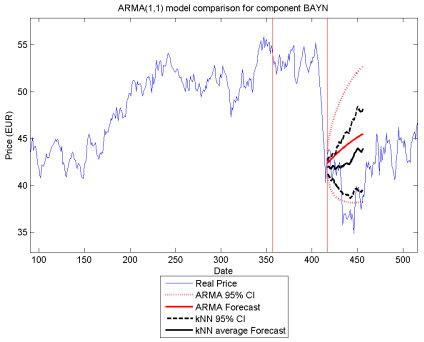
(C) 50% "history" BAS



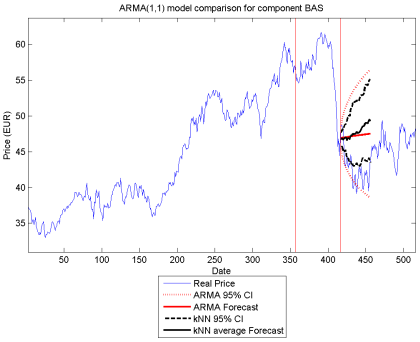
(D) 50% "history" BAYN



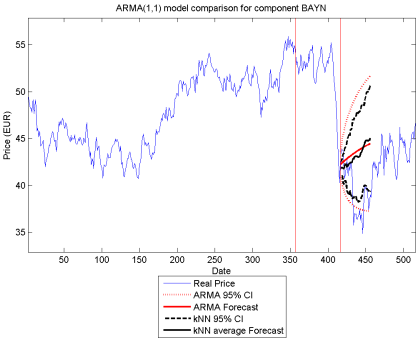
(E) 75% "history" BAS



(F) 75% "history" BAYN

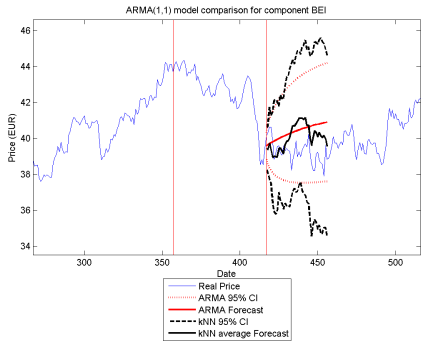


(G) 100% "history" BAS

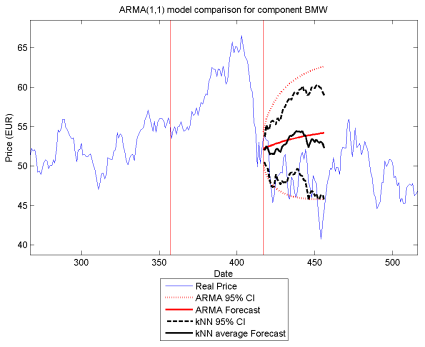


(H) 100% "history" BAYN

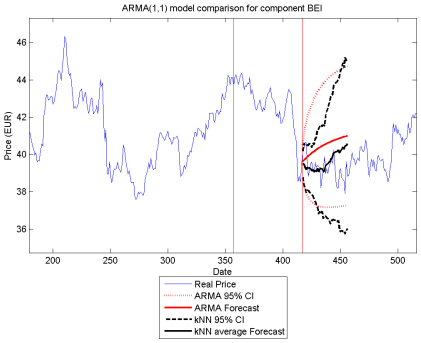
FIGURE B.2: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component BAS (all LEFT figures) and BAYN (all RIGHT figures)



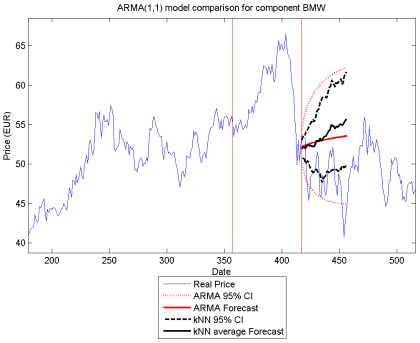
(A) 25% "history" BEI



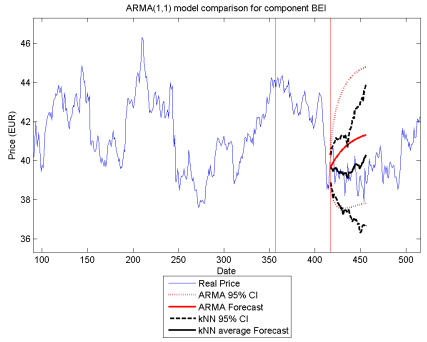
(B) 25% "history" BMW



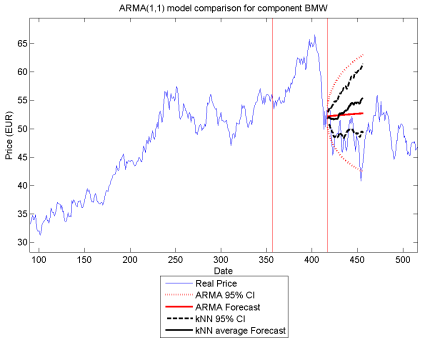
(C) 50% "history" BEI



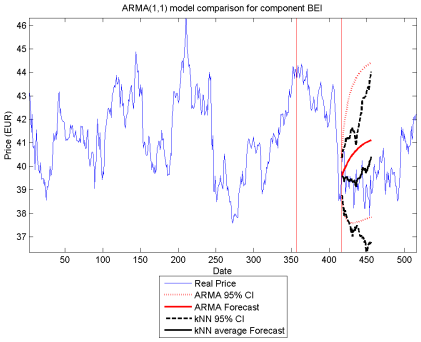
(D) 50% "history" BMW



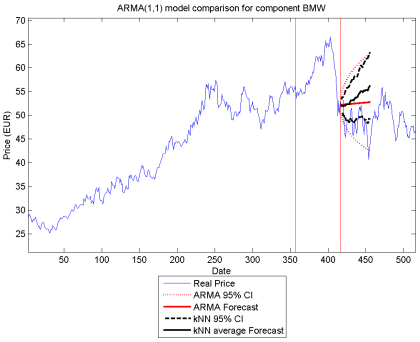
(E) 75% "history" BEI



(F) 75% "history" BMW

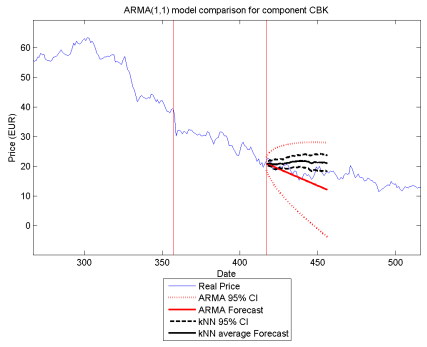


(G) 100% "history" BEI

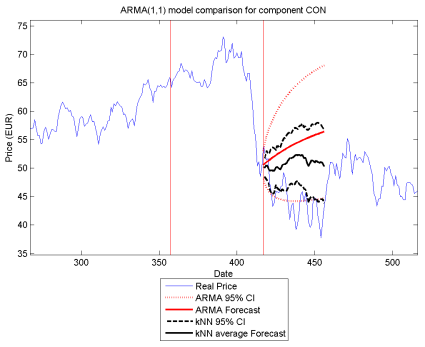


(H) 100% "history" BMW

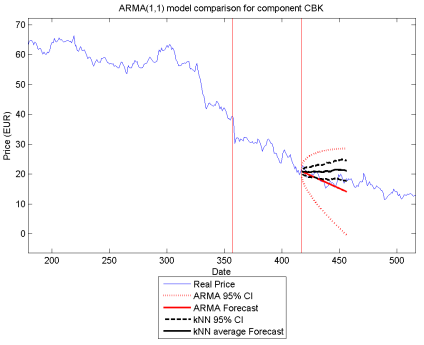
FIGURE B.3: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component BEI (all LEFT figures) and BMW (all RIGHT figures)



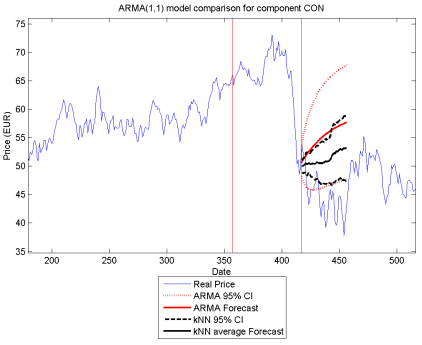
(A) 25% "history" CBK



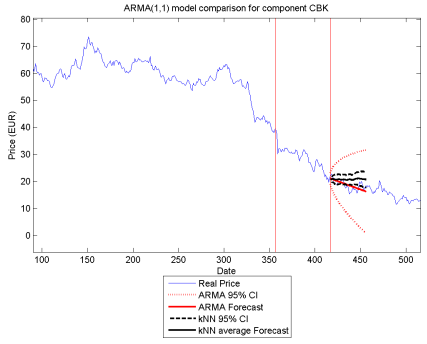
(B) 25% "history" CON



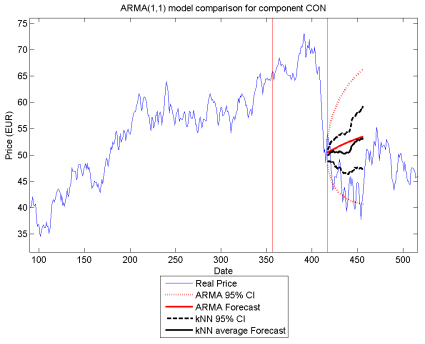
(C) 50% "history" CBK



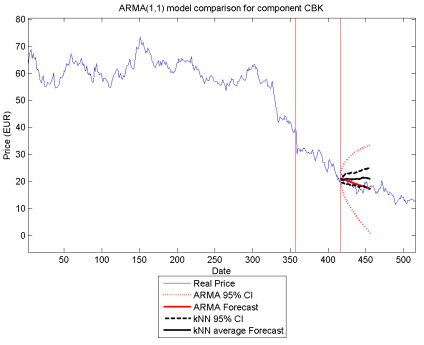
(D) 50% "history" CON



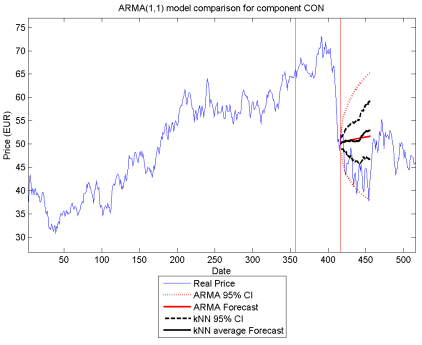
(E) 75% "history" CBK



(F) 75% "history" CON

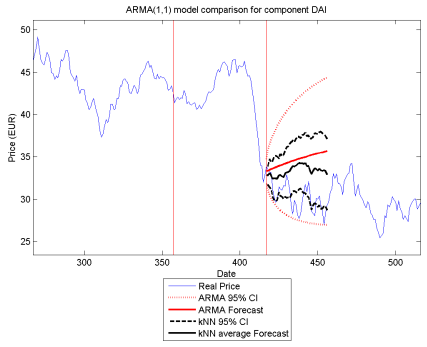


(G) 100% "history" CBK

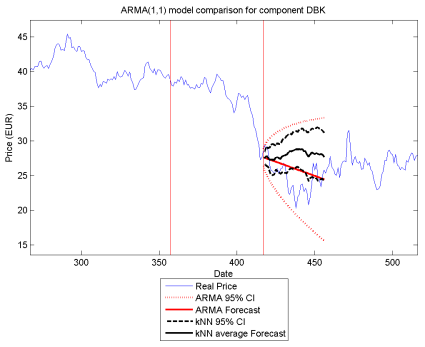


(H) 100% "history" CON

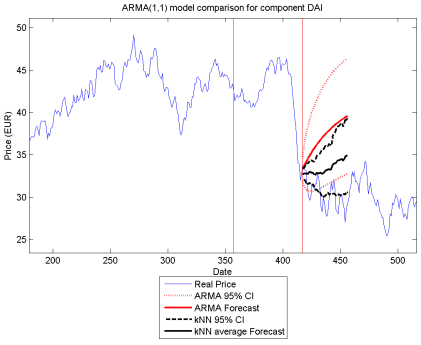
FIGURE B.4: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component CBK (all LEFT figures) and CON (all RIGHT figures)



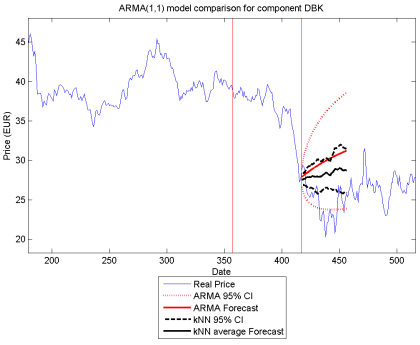
(A) 25% "history" DAI



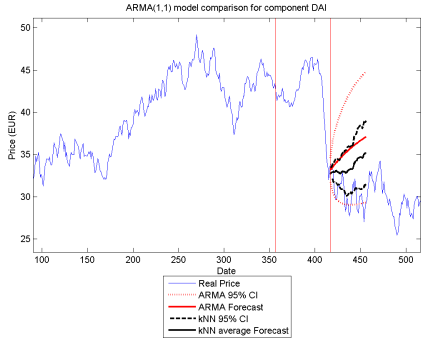
(B) 25% "history" DBK



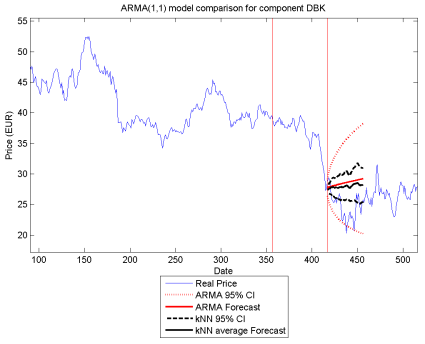
(C) 50% "history" DAI



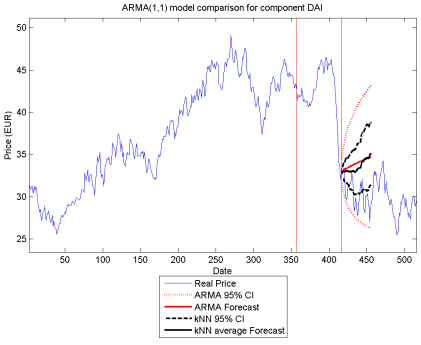
(D) 50% "history" DBK



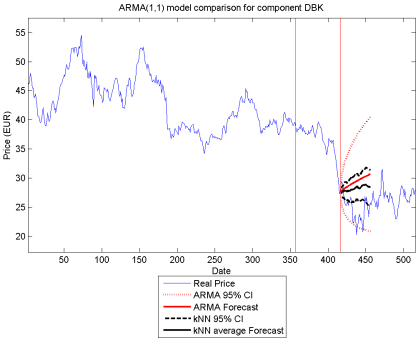
(E) 75% "history" DAI



(F) 75% "history" DBK

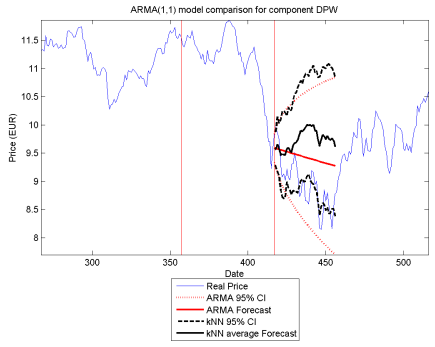


(G) 100% "history" DAI

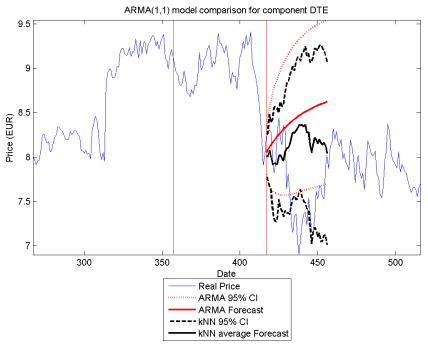


(H) 100% "history" DBK

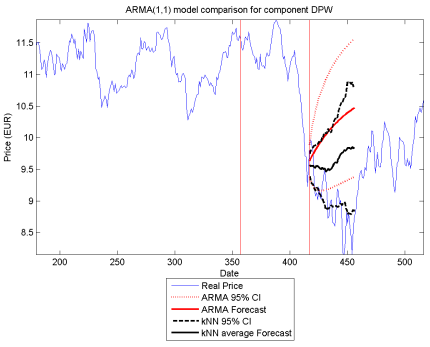
FIGURE B.5: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component DAI (all LEFT figures) and DBK (all RIGHT figures)



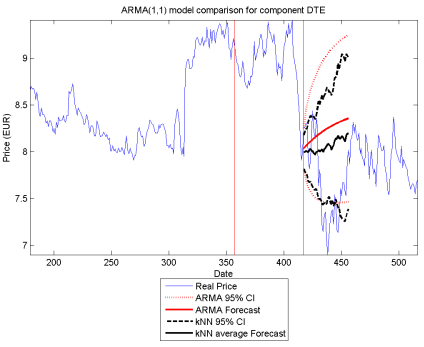
(A) 25% "history" DPW



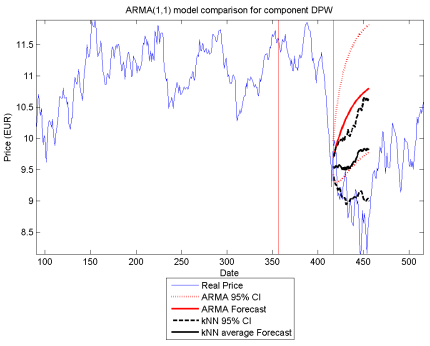
(B) 25% "history" DTE



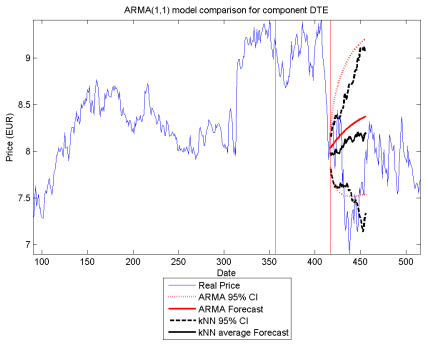
(C) 50% "history" DPW



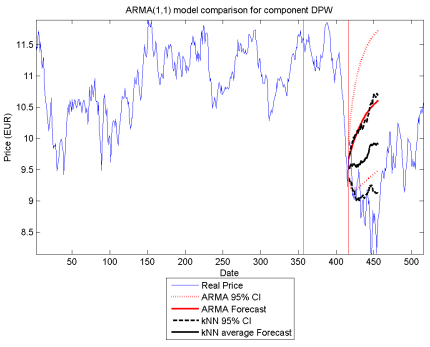
(D) 50% "history" DTE



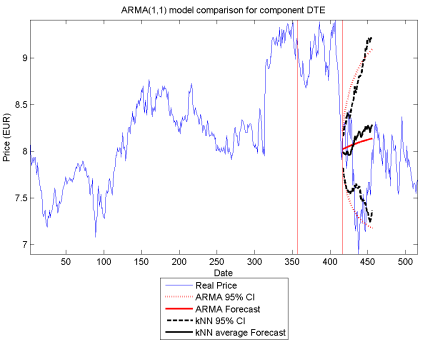
(E) 75% "history" DPW



(F) 75% "history" DTE

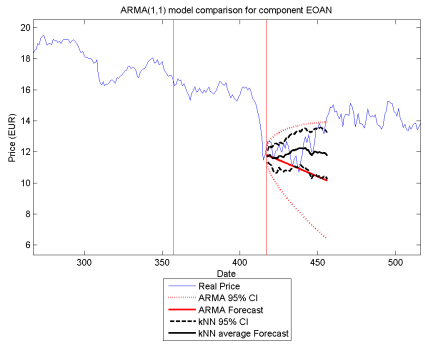


(G) 100% "history" DPW

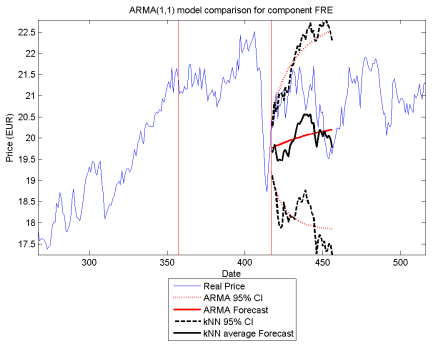


(H) 100% "history" DTE

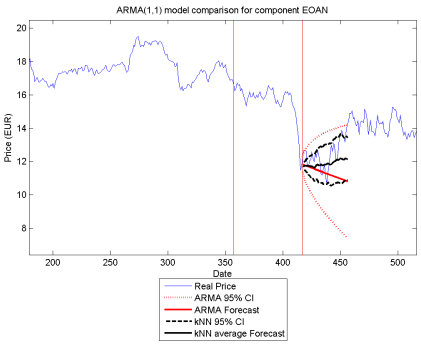
FIGURE B.6: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component DPW (all LEFT figures) and DTE (all RIGHT figures)



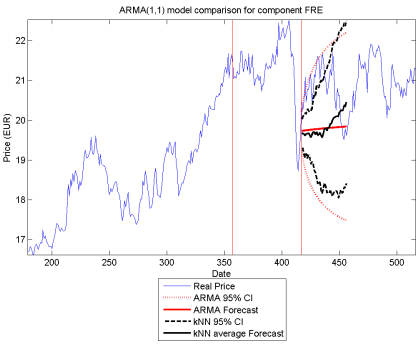
(A) 25% "history" EOAN



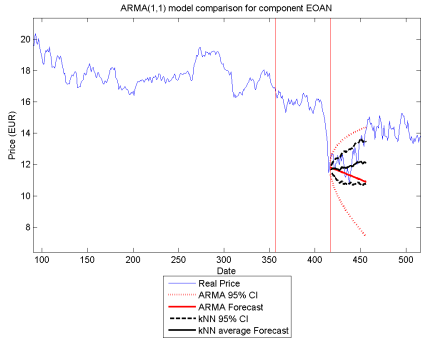
(B) 25% "history" FRE



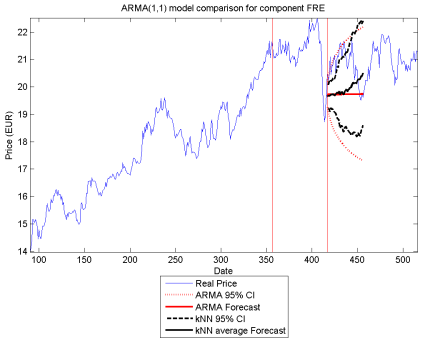
(C) 50% "history" EOAN



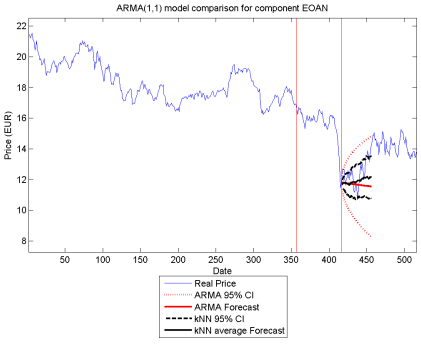
(D) 50% "history" FRE



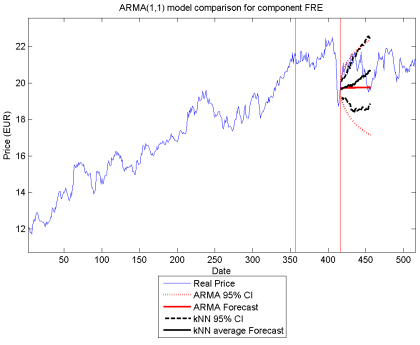
(E) 75% "history" EOAN



(F) 75% "history" FRE

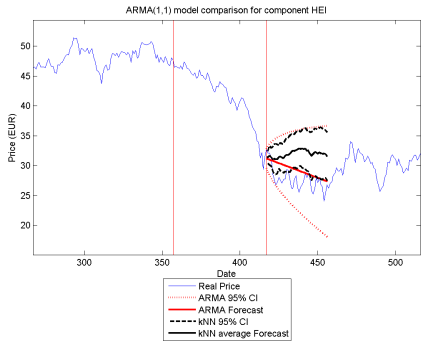


(G) 100% "history" EOAN

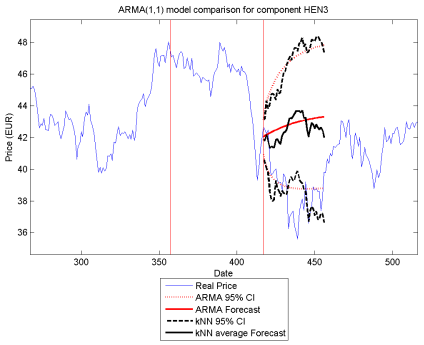


(H) 100% "history" FRE

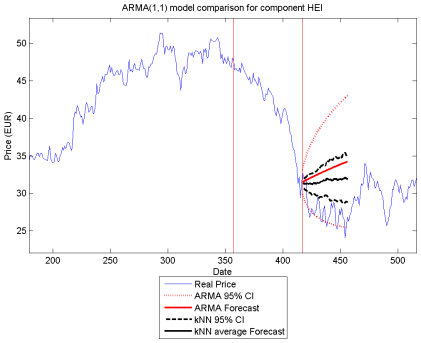
FIGURE B.7: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component EOAN (all LEFT figures) and FRE (all RIGHT figures)



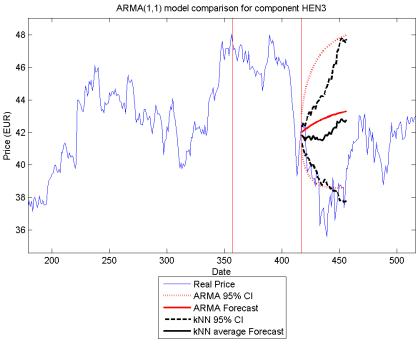
(A) 25% "history" HEI



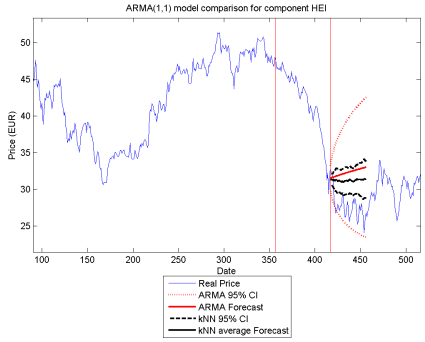
(B) 25% "history" HEN3



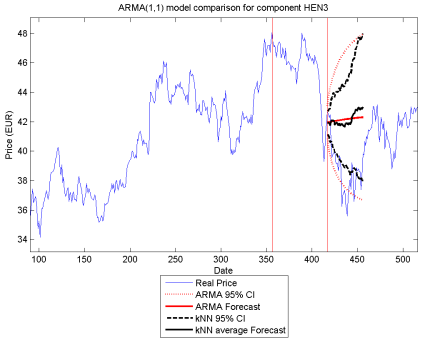
(C) 50% "history" HEI



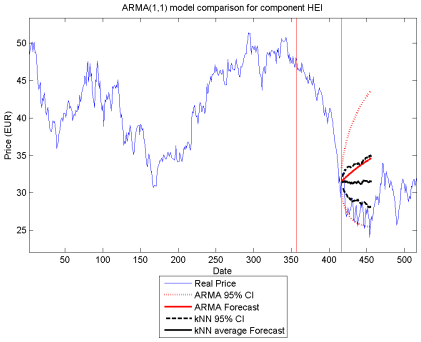
(D) 50% "history" HEN3



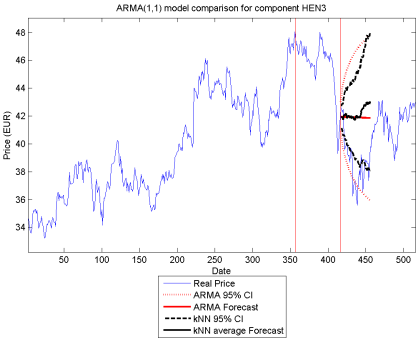
(E) 75% "history" HEI



(F) 75% "history" HEN3

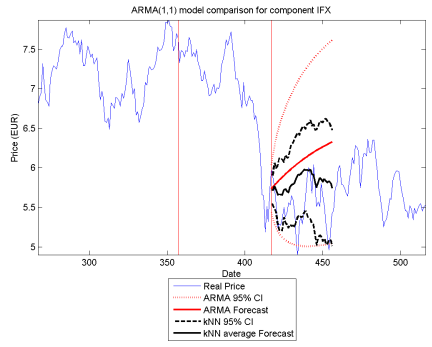


(G) 100% "history" HEI

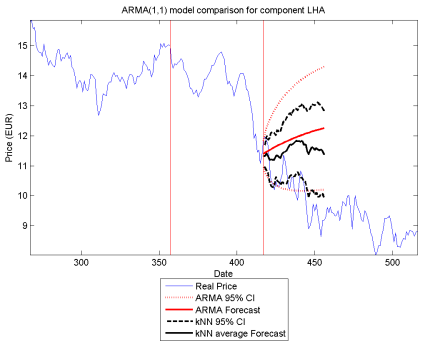


(H) 100% "history" HEN3

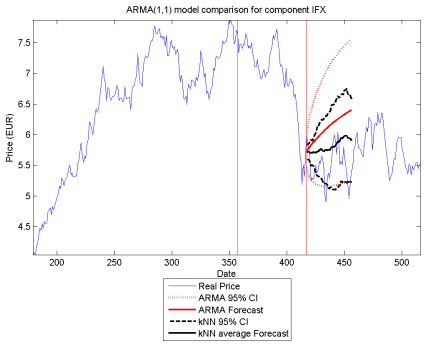
FIGURE B.8: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component HEI (all LEFT figures) and HEN3 (all RIGHT figures)



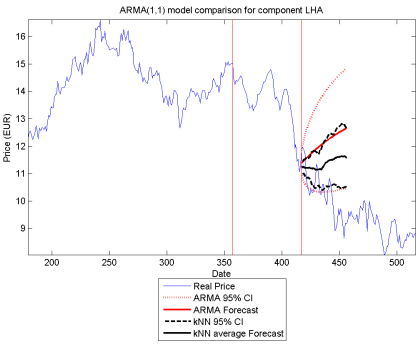
(A) 25% "history" IFX



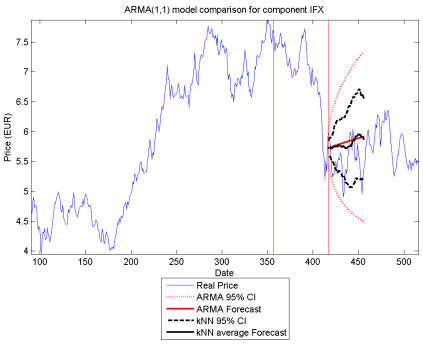
(B) 25% "history" LHA



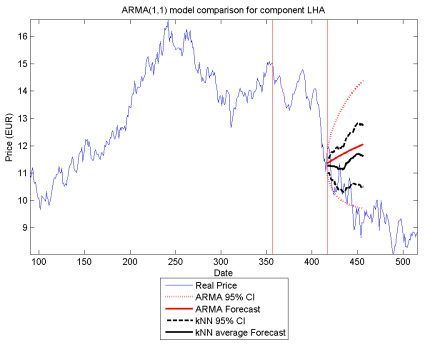
(C) 50% "history" IFX



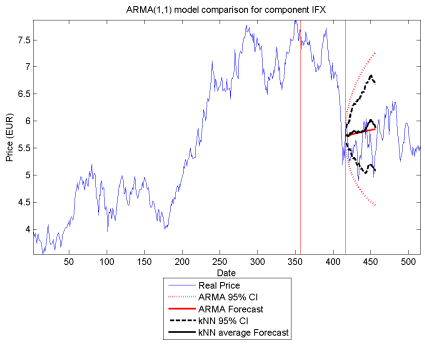
(D) 50% "history" LHA



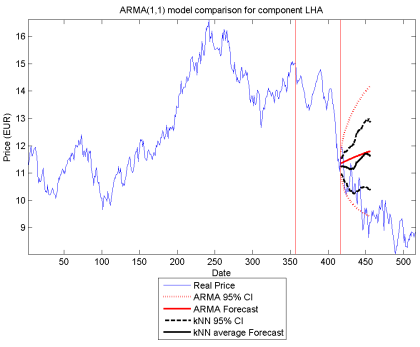
(E) 75% "history" IFX



(F) 75% "history" LHA

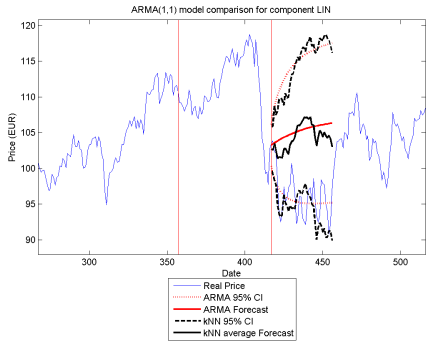


(G) 100% "history" IFX

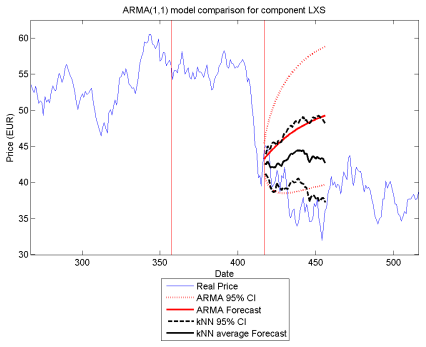


(H) 100% "history" LHA

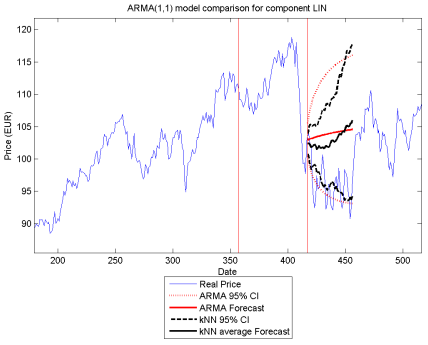
FIGURE B.9: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component IFX (all LEFT figures) and LHA (all RIGHT figures)



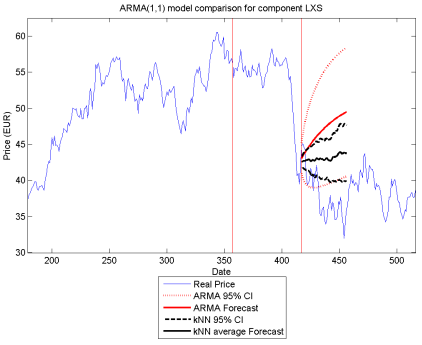
(A) 25% "history" LIN



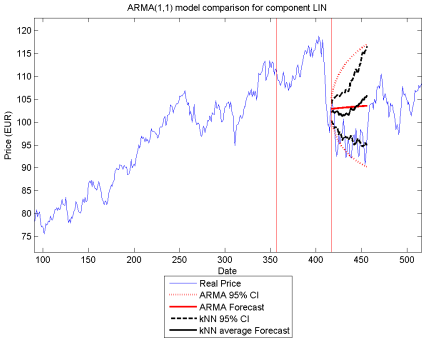
(B) 25% "history" LXS



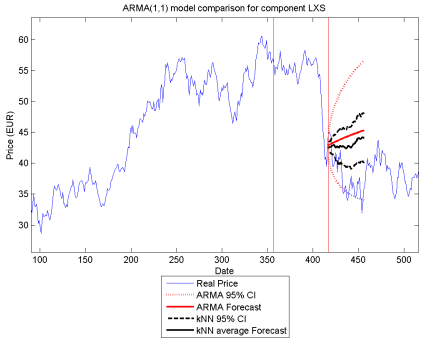
(C) 50% "history" LIN



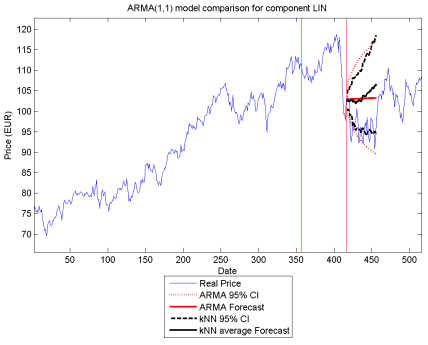
(D) 50% "history" LXS



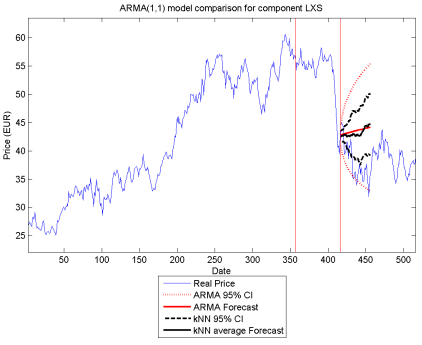
(E) 75% "history" LIN



(F) 75% "history" LXS

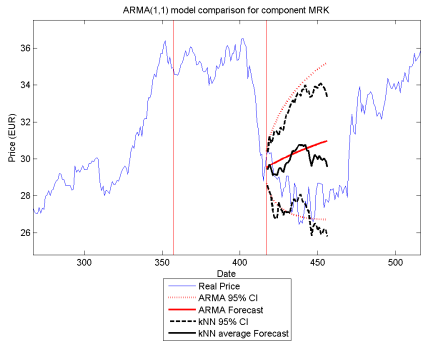


(G) 100% "history" LIN

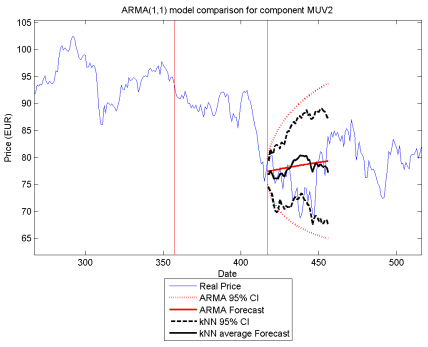


(H) 100% "history" LXS

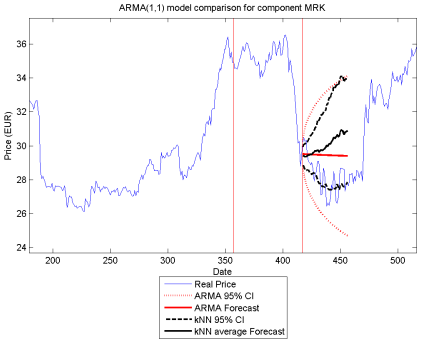
FIGURE B.10: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component LIN (all LEFT figures) and LXS (all RIGHT figures)



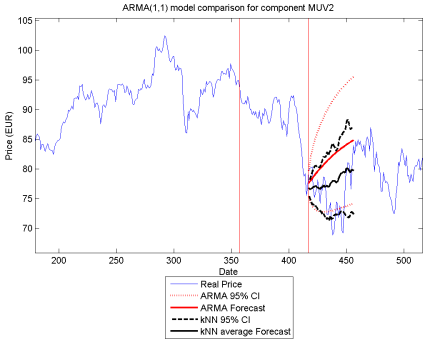
(A) 25% "history" MRK



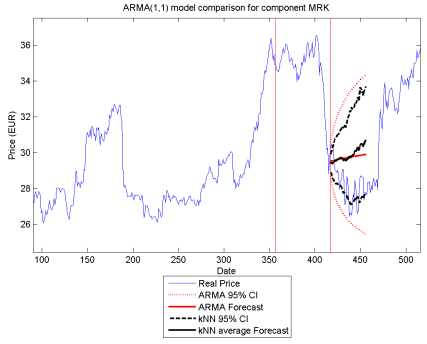
(B) 25% "history" MUV2



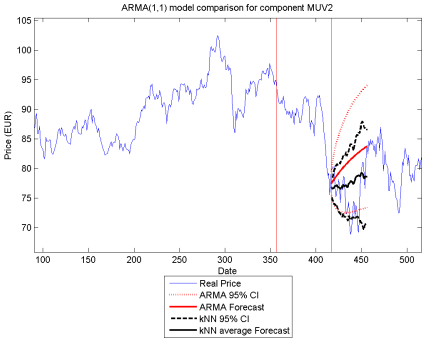
(C) 50% "history" MRK



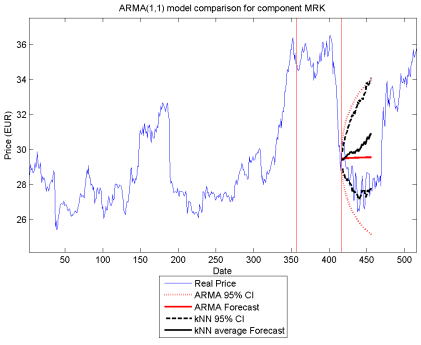
(D) 50% "history" MUV2



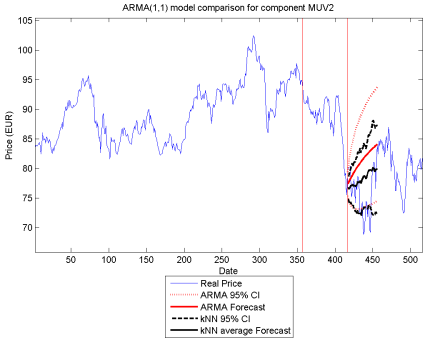
(E) 75% "history" MRK



(F) 75% "history" MUV2

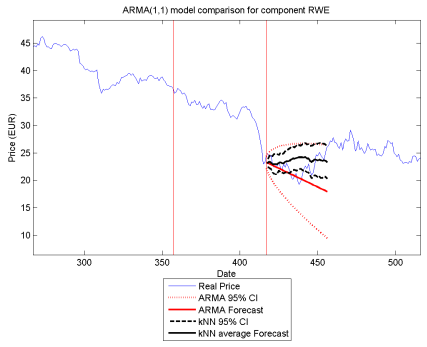


(G) 100% "history" MRK

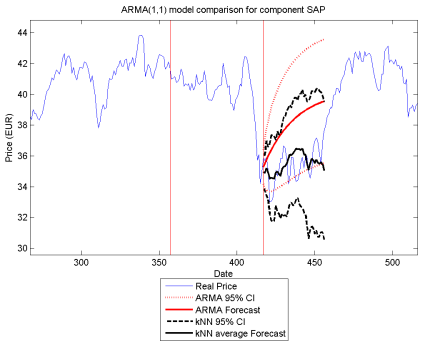


(H) 100% "history" MUV2

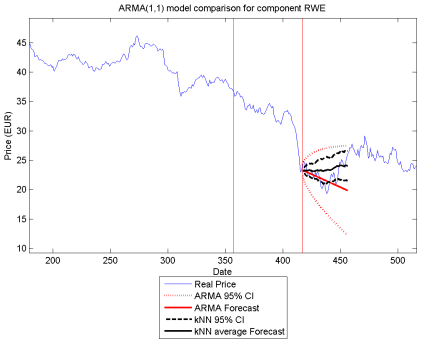
FIGURE B.11: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component MRK (all LEFT figures) and MUV2 (all RIGHT figures)



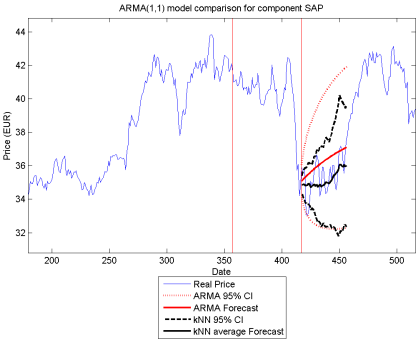
(A) 25% "history" RWE



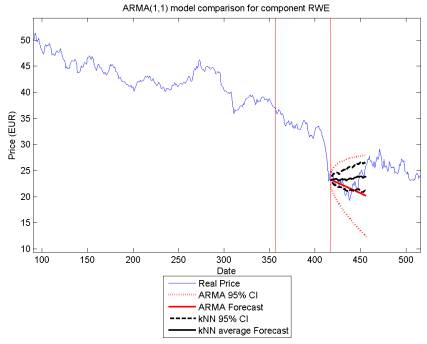
(B) 25% "history" SAP



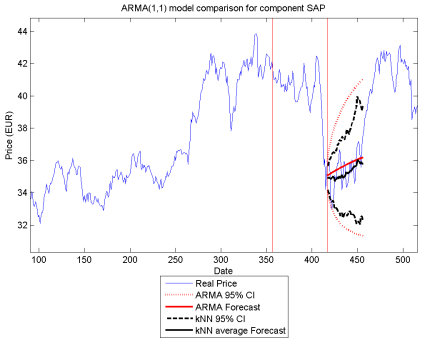
(C) 50% "history" RWE



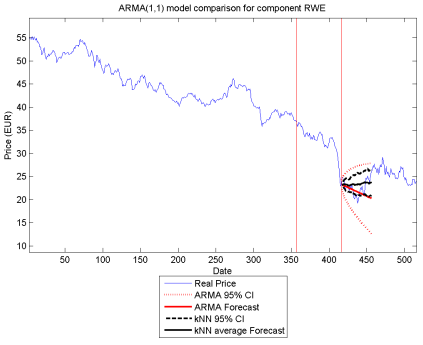
(D) 50% "history" SAP



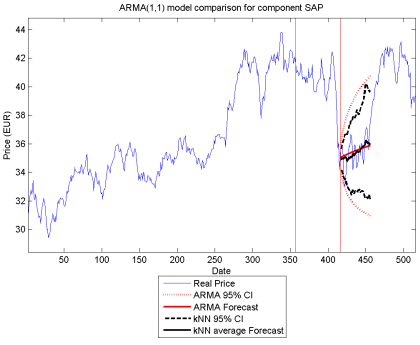
(E) 75% "history" RWE



(F) 75% "history" SAP

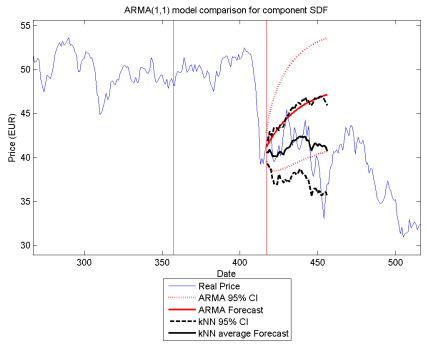


(G) 100% "history" RWE

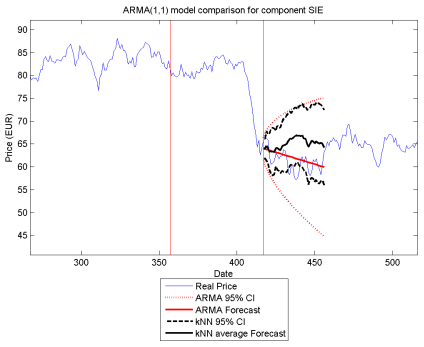


(H) 100% "history" SAP

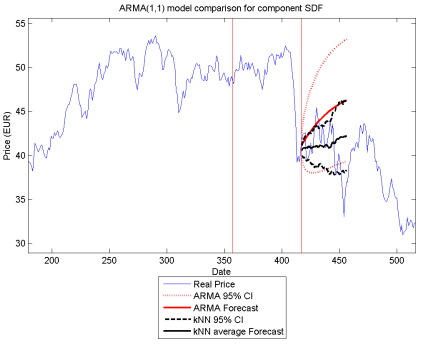
FIGURE B.12: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component RWE (all LEFT figures) and SAP (all RIGHT figures)



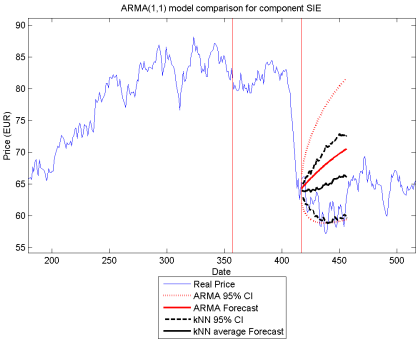
(A) 25% "history" SDF



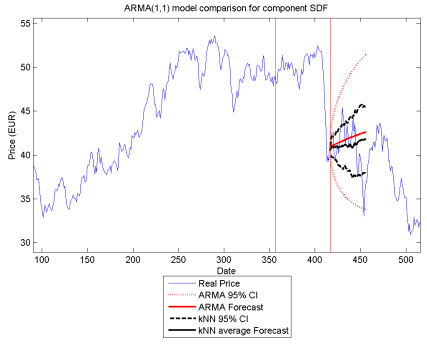
(B) 25% "history" SIE



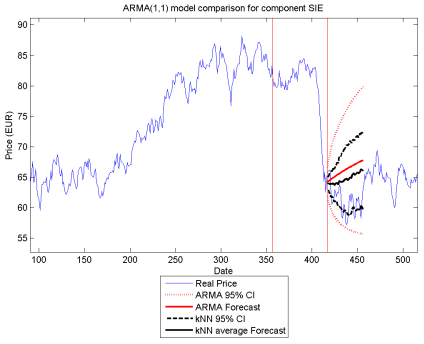
(C) 50% "history" SDF



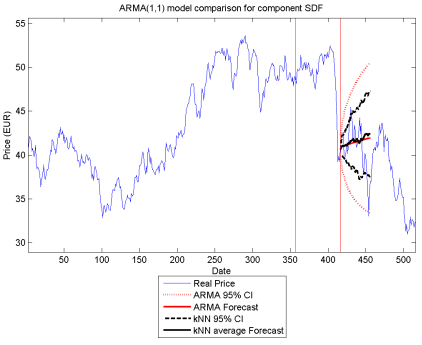
(D) 50% "history" SIE



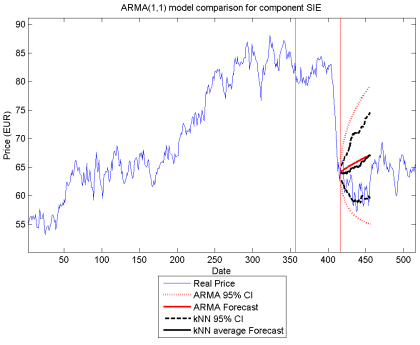
(E) 75% "history" SDF



(F) 75% "history" SIE

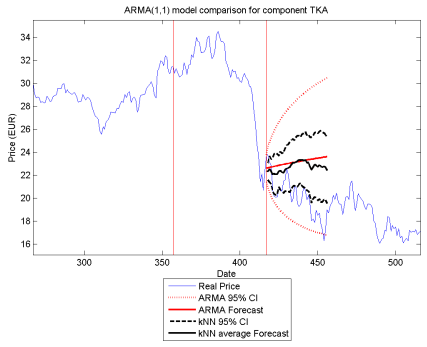


(G) 100% "history" SDF

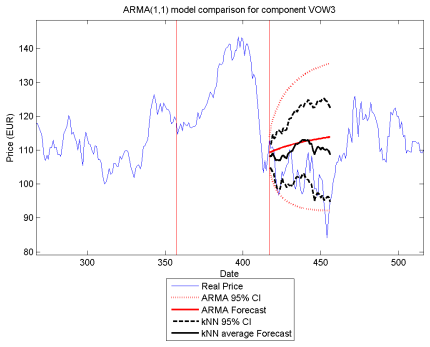


(H) 100% "history" SIE

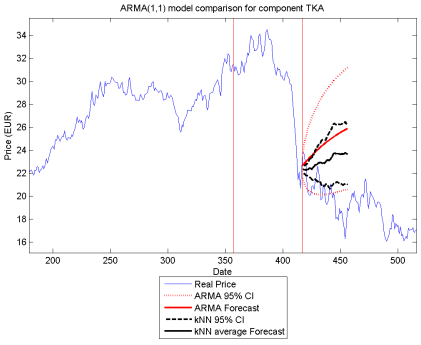
FIGURE B.13: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component SDF (all LEFT figures) and SIE (all RIGHT figures)



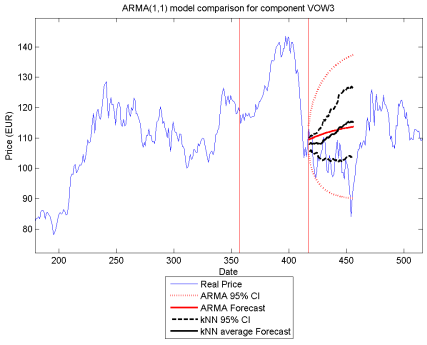
(A) 25% "history" TKA



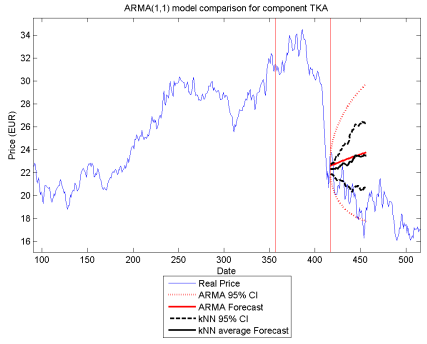
(B) 25% "history" VOW3



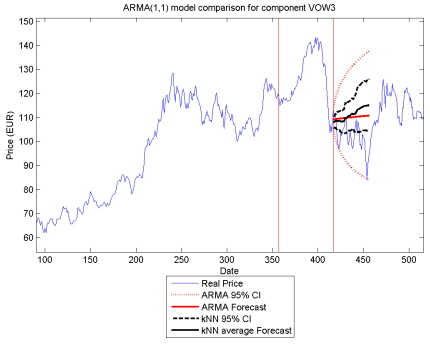
(C) 50% "history" TKA



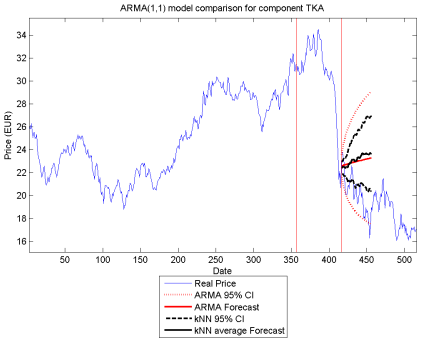
(D) 50% "history" VOW3



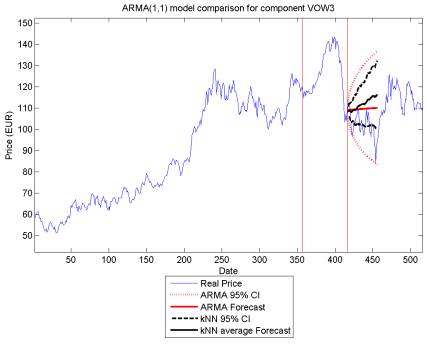
(E) 75% "history" TKA



(F) 75% "history" VOW3



(G) 100% "history" TKA

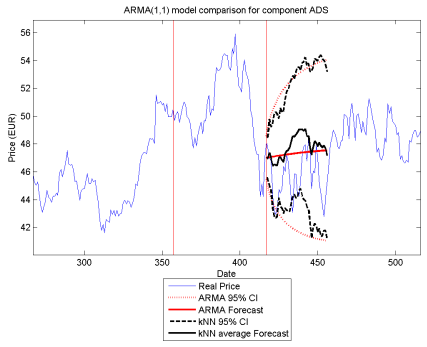


(H) 100% "history" VOW3

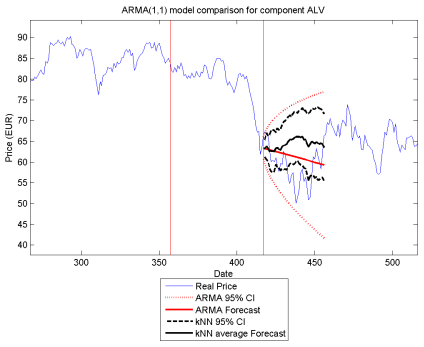
FIGURE B.14: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (City Block Distance) for component TKA (all LEFT figures) and VOW3 (all RIGHT figures)

Appendix C

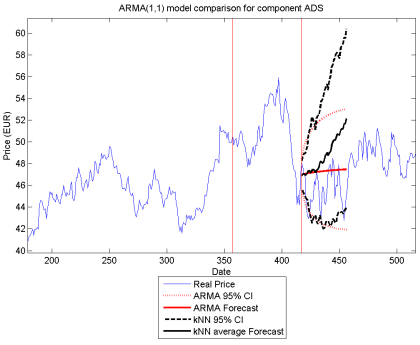
Graphs of comparison between the kNN approach and the ARMA(1,1) model (Correlation Distance)



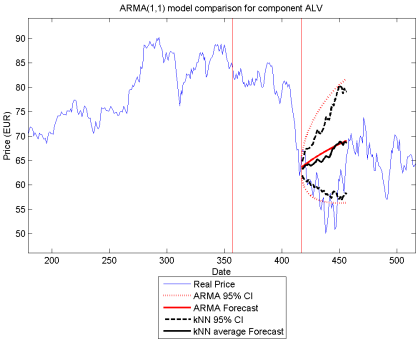
(A) 25% "history" ADS



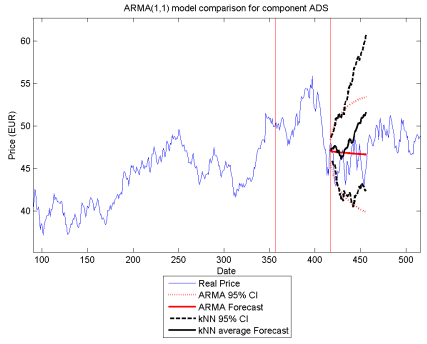
(B) 25% "history" ALV



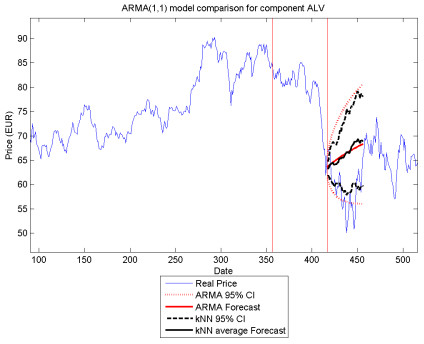
(C) 50% "history" ADS



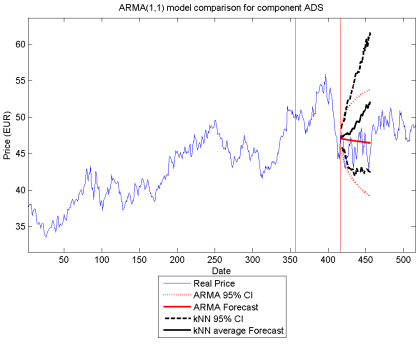
(D) 50% "history" ALV



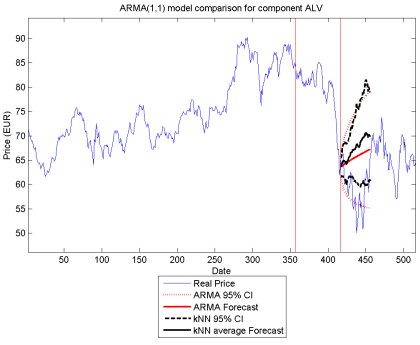
(E) 75% "history" ADS



(F) 75% "history" ALV

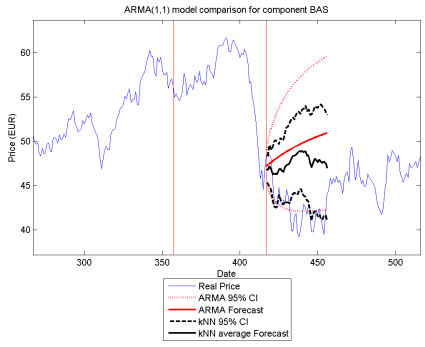


(G) 100% "history" ADS

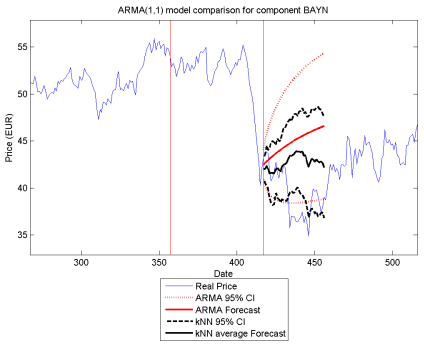


(H) 100% "history" ALV

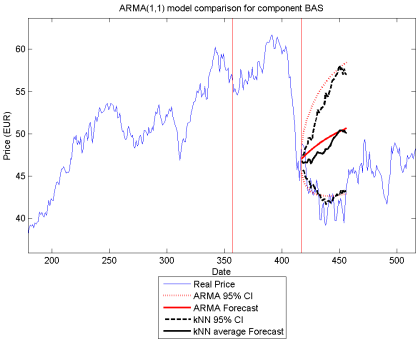
FIGURE C.1: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component ADS (all LEFT figures) and ALV (all RIGHT figures)



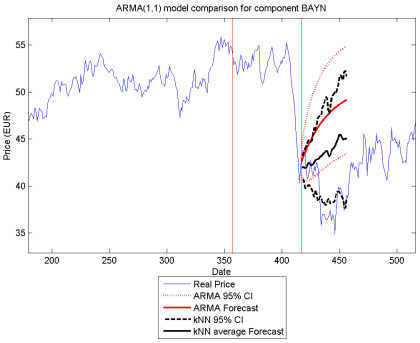
(A) 25% "history" BAS



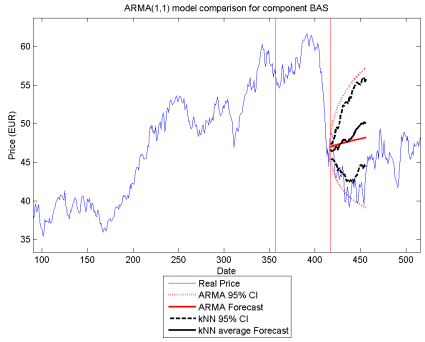
(B) 25% "history" BAYN



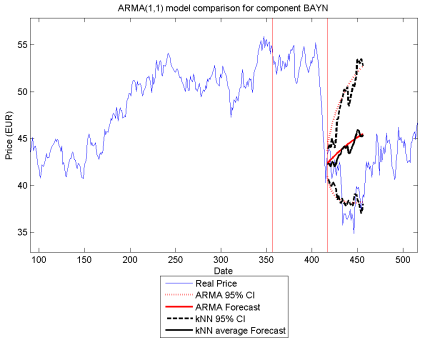
(C) 50% "history" BAS



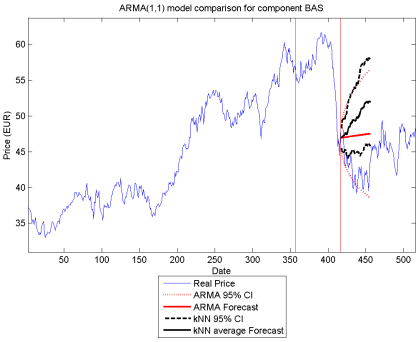
(D) 50% "history" BAYN



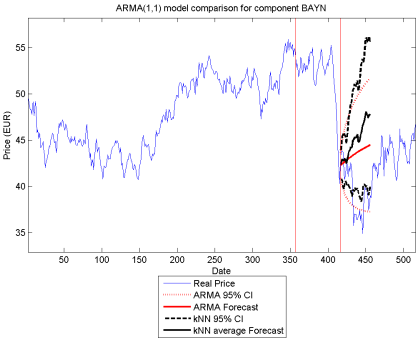
(E) 75% "history" BAS



(F) 75% "history" BAYN

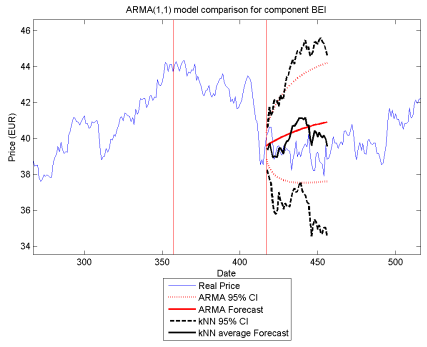


(G) 100% "history" BAS

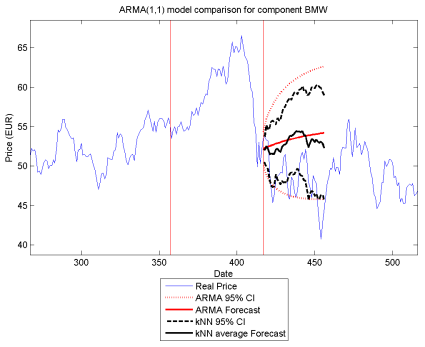


(H) 100% "history" BAYN

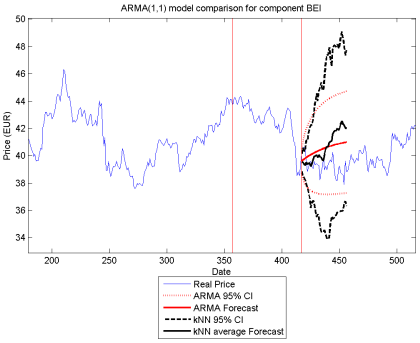
FIGURE C.2: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component BAS (all LEFT figures) and BAYN (all RIGHT figures)



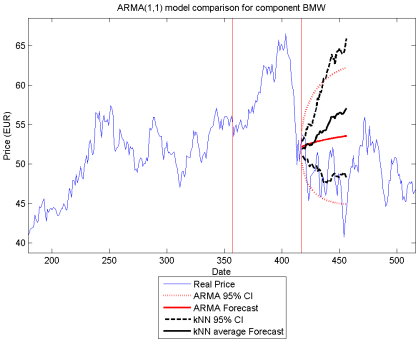
(A) 25% "history" BEI



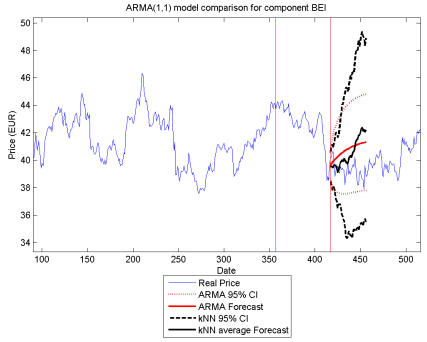
(B) 25% "history" BMW



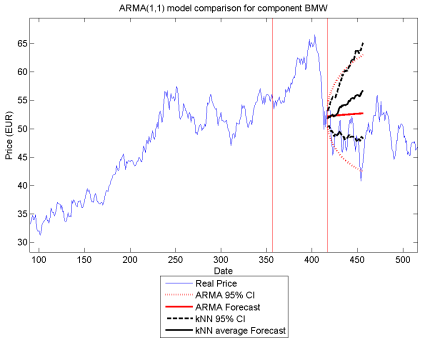
(C) 50% "history" BEI



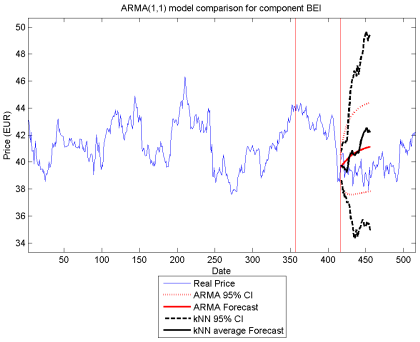
(D) 50% "history" BMW



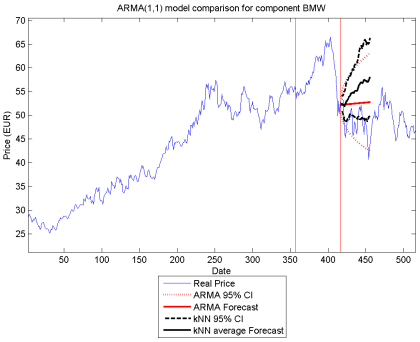
(E) 75% "history" BEI



(F) 75% "history" BMW

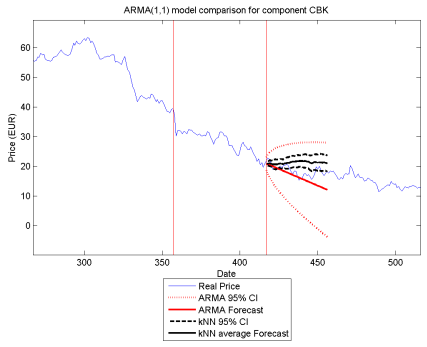


(G) 100% "history" BEI

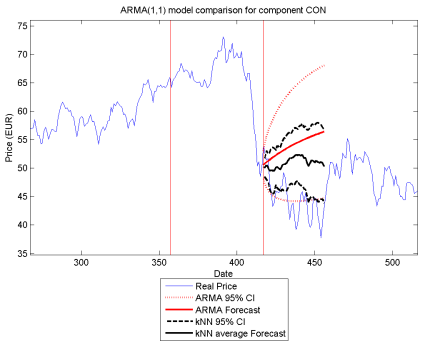


(H) 100% "history" BMW

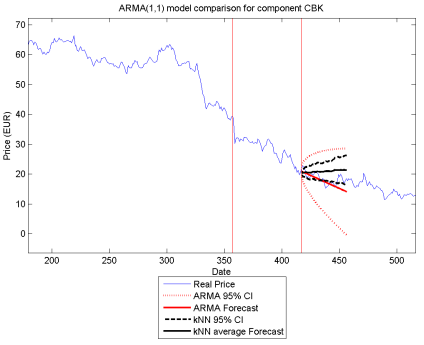
FIGURE C.3: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component BEI (all LEFT figures) and BMW (all RIGHT figures)



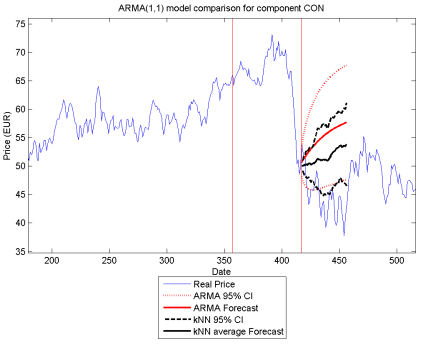
(A) 25% "history" CBK



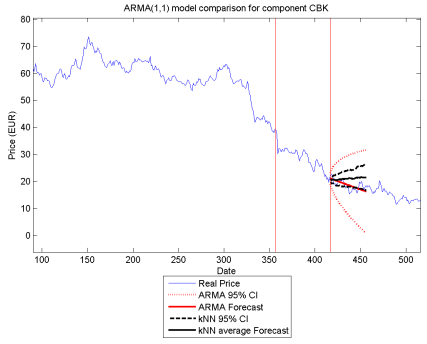
(B) 25% "history" CON



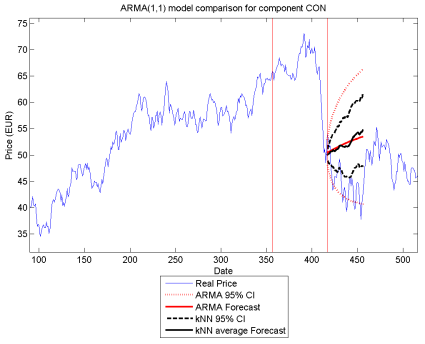
(C) 50% "history" CBK



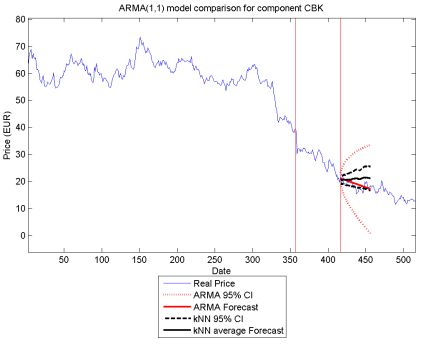
(D) 50% "history" CON



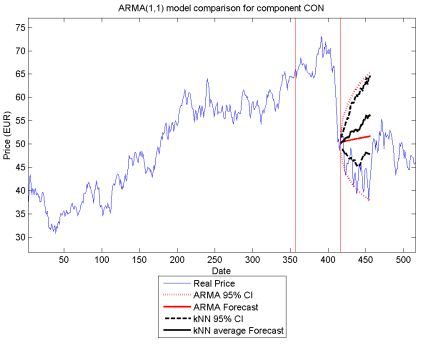
(E) 75% "history" CBK



(F) 75% "history" CON

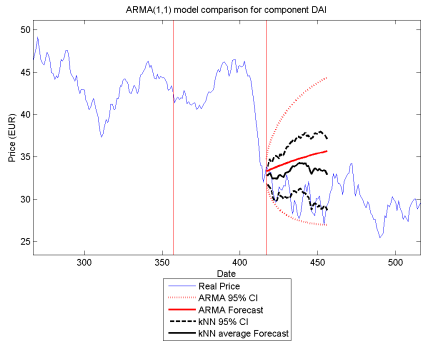


(G) 100% "history" CBK

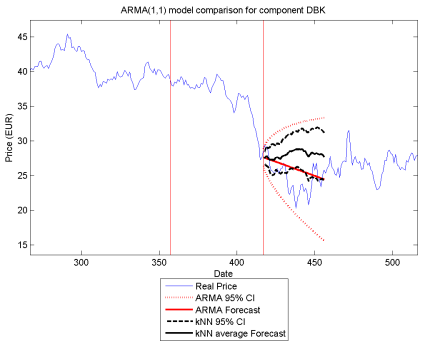


(H) 100% "history" CON

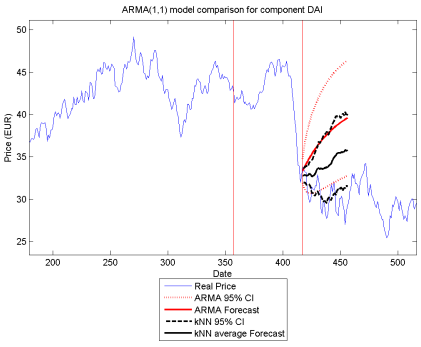
FIGURE C.4: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component CBK (all LEFT figures) and CON (all RIGHT figures)



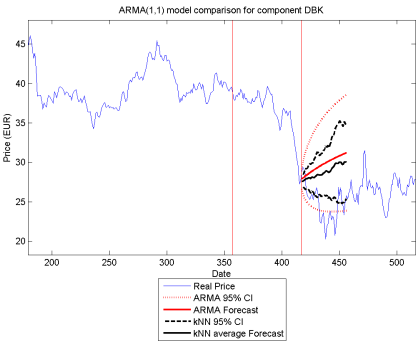
(A) 25% "history" DAI



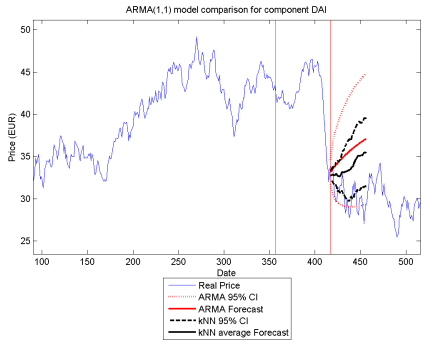
(B) 25% "history" DBK



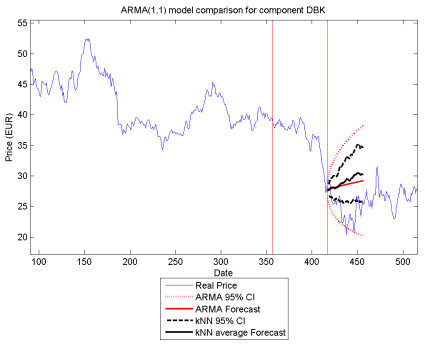
(C) 50% "history" DAI



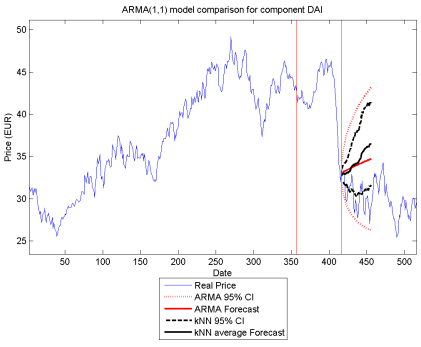
(D) 50% "history" DBK



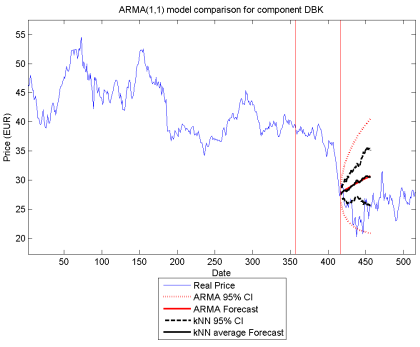
(E) 75% "history" DAI



(F) 75% "history" DBK

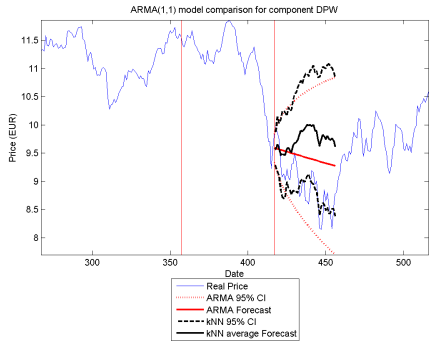


(G) 100% "history" DAI

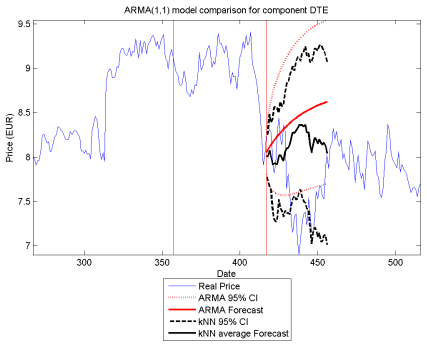


(H) 100% "history" DBK

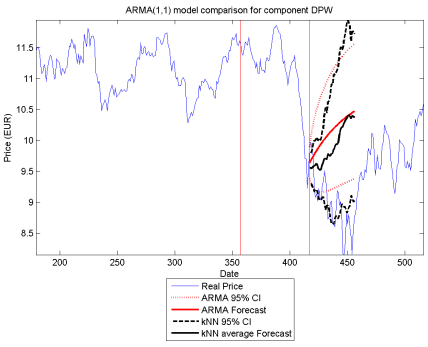
FIGURE C.5: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component DAI (all LEFT figures) and DBK (all RIGHT figures)



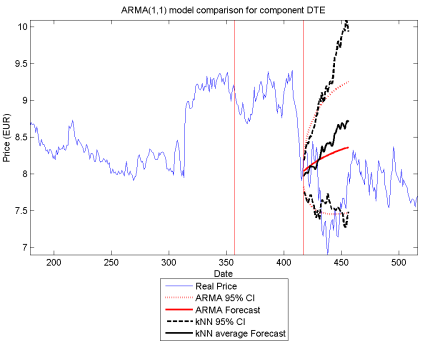
(A) 25% "history" DPW



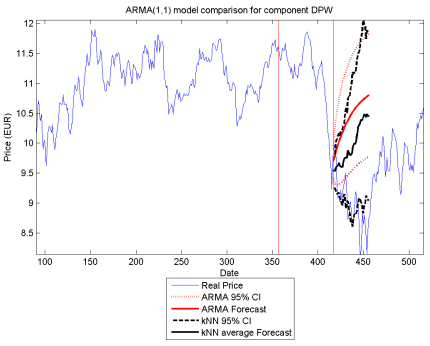
(B) 25% "history" DTE



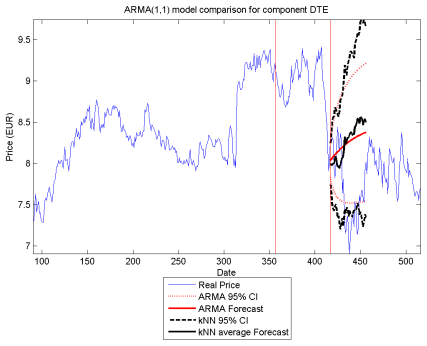
(C) 50% "history" DPW



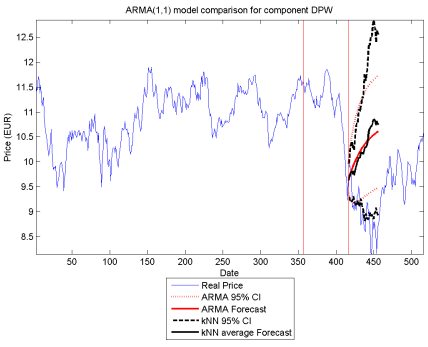
(D) 50% "history" DTE



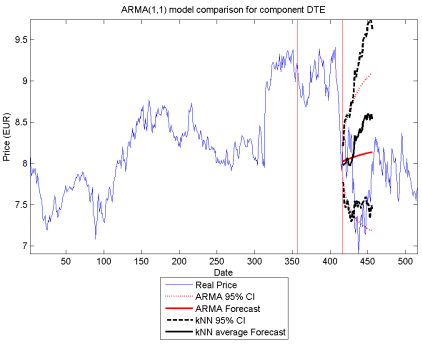
(E) 75% "history" DPW



(F) 75% "history" DTE

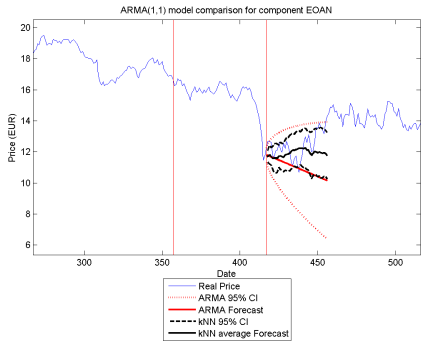


(G) 100% "history" DPW

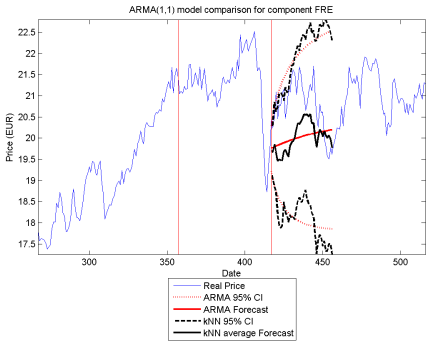


(H) 100% "history" DTE

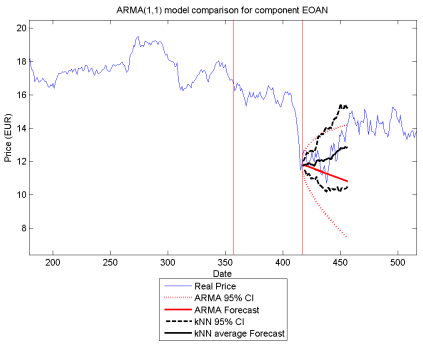
FIGURE C.6: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component DPW (all LEFT figures) and DTE (all RIGHT figures)



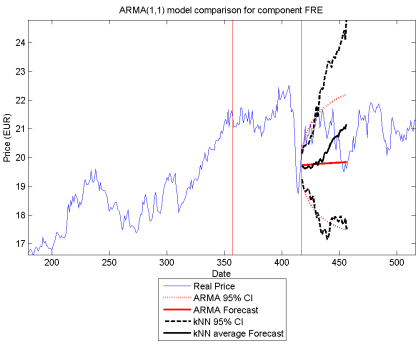
(A) 25% "history" EOAN



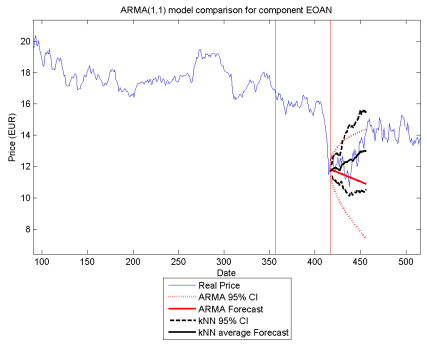
(B) 25% "history" FRE



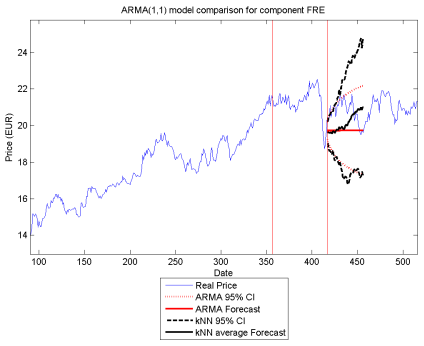
(C) 50% "history" EOAN



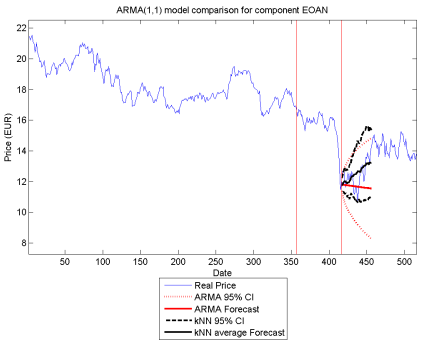
(D) 50% "history" FRE



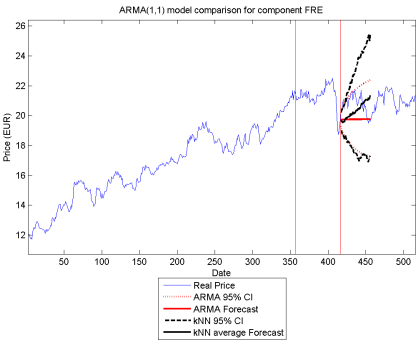
(E) 75% "history" EOAN



(F) 75% "history" FRE

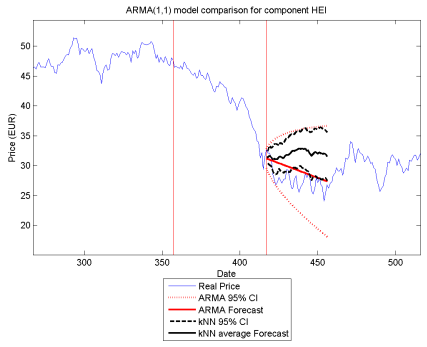


(G) 100% "history" EOAN

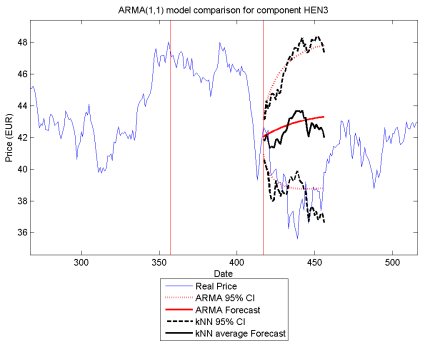


(H) 100% "history" FRE

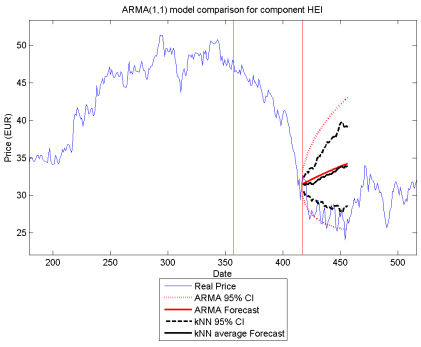
FIGURE C.7: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component EOAN (all LEFT figures) and FRE (all RIGHT figures)



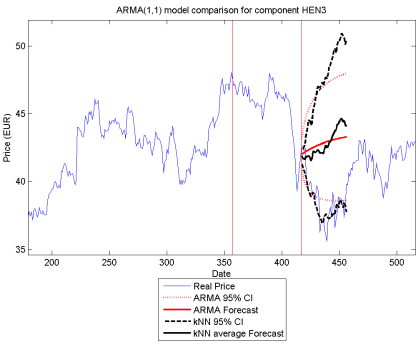
(A) 25% "history" HEI



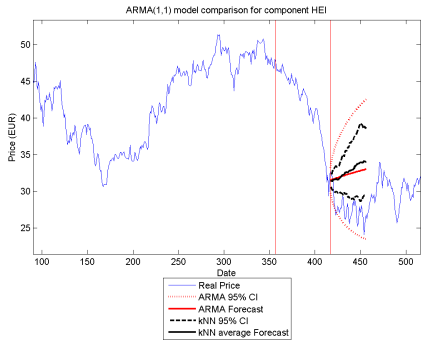
(B) 25% "history" HEN3



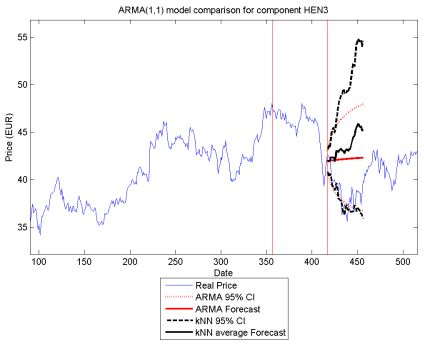
(C) 50% "history" HEI



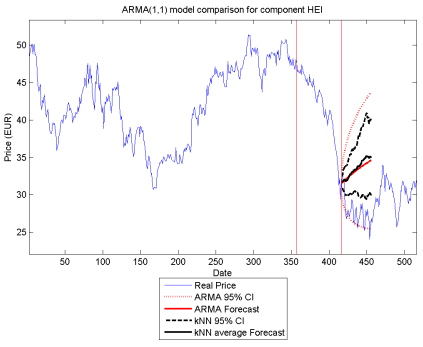
(D) 50% "history" HEN3



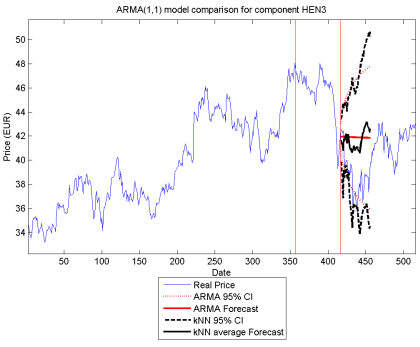
(E) 75% "history" HEI



(F) 75% "history" HEN3

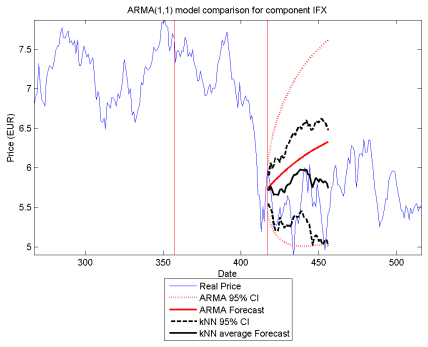


(G) 100% "history" HEI

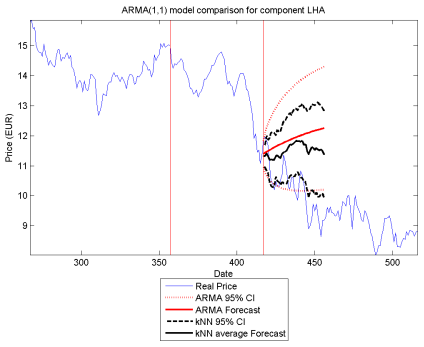


(H) 100% "history" HEN3

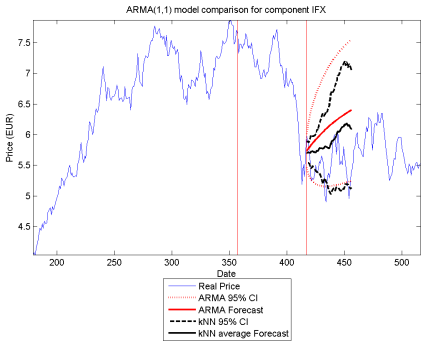
FIGURE C.8: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component HEI (all LEFT figures) and HEN3 (all RIGHT figures)



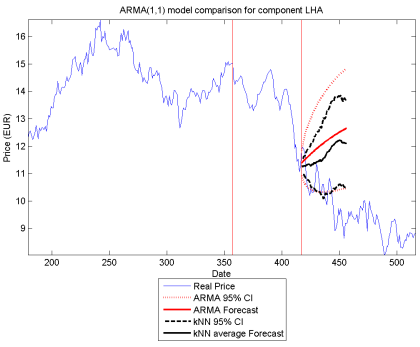
(A) 25% "history" IFX



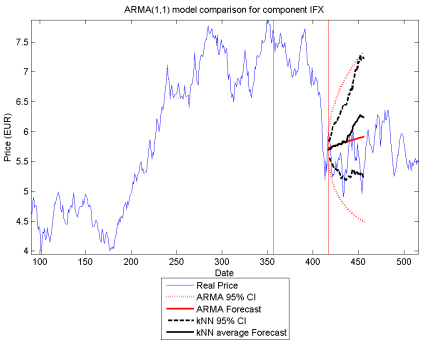
(B) 25% "history" LHA



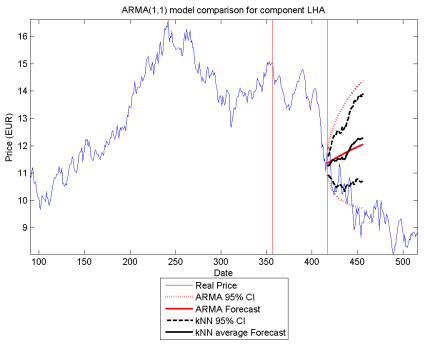
(C) 50% "history" IFX



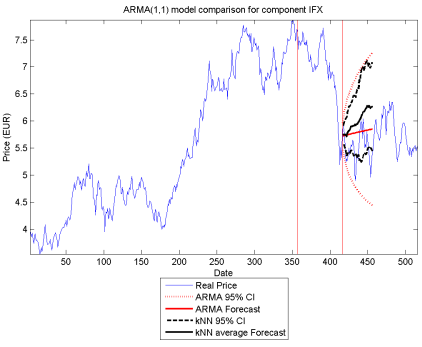
(D) 50% "history" LHA



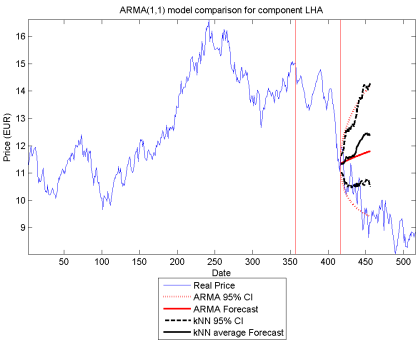
(E) 75% "history" IFX



(F) 75% "history" LHA

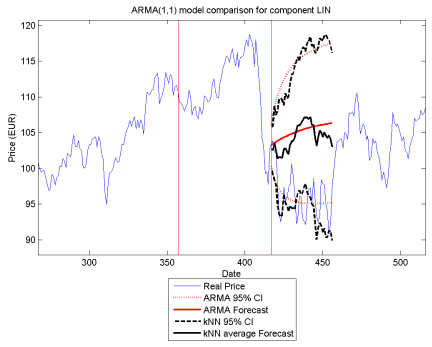


(G) 100% "history" IFX

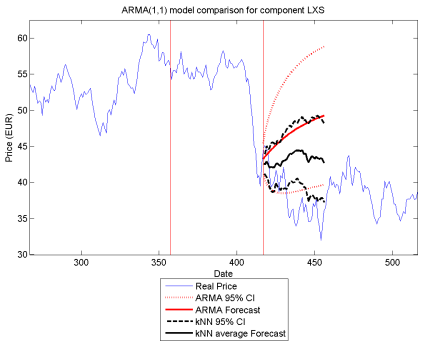


(H) 100% "history" LHA

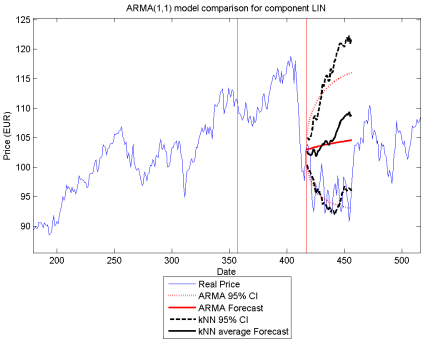
FIGURE C.9: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component IFX (all LEFT figures) and LHA (all RIGHT figures)



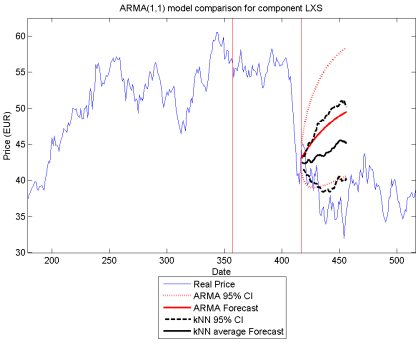
(A) 25% "history" LIN



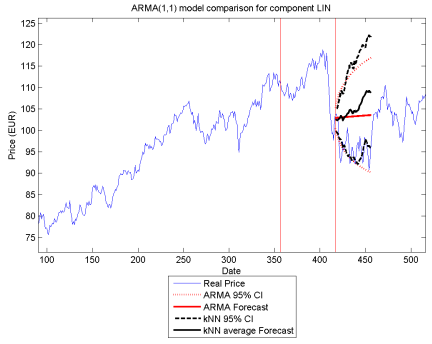
(B) 25% "history" LXS



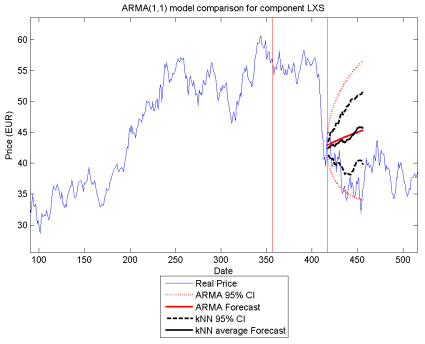
(C) 50% "history" LIN



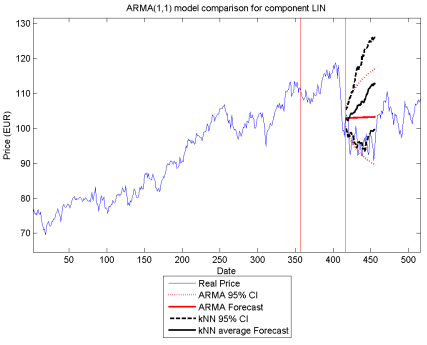
(D) 50% "history" LXS



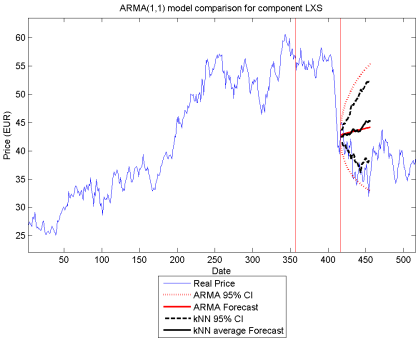
(E) 75% "history" LIN



(F) 75% "history" LXS

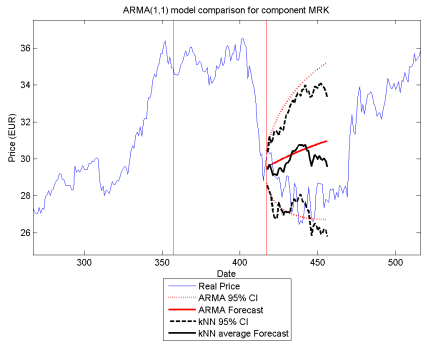


(G) 100% "history" LIN

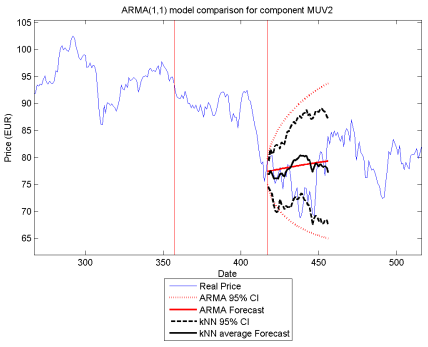


(H) 100% "history" LXS

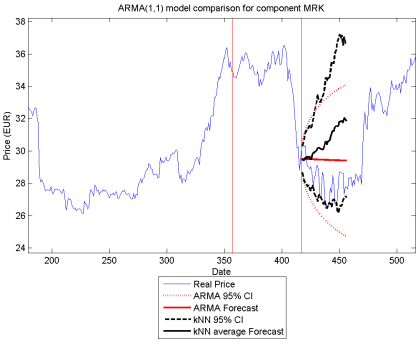
FIGURE C.10: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component LIN (all LEFT figures) and LXS (all RIGHT figures)



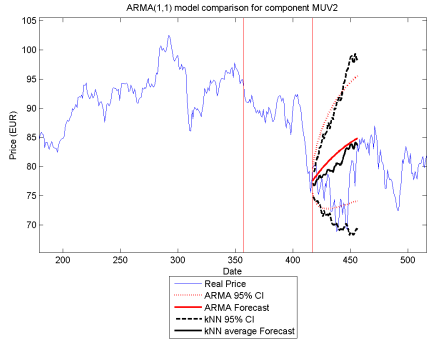
(A) 25% "history" MRK



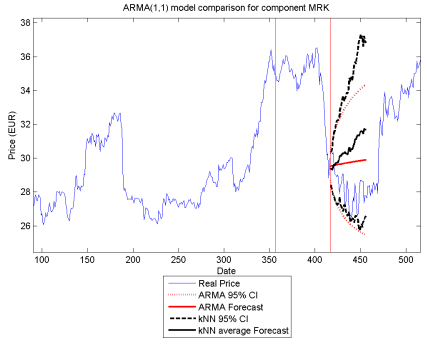
(B) 25% "history" MUV2



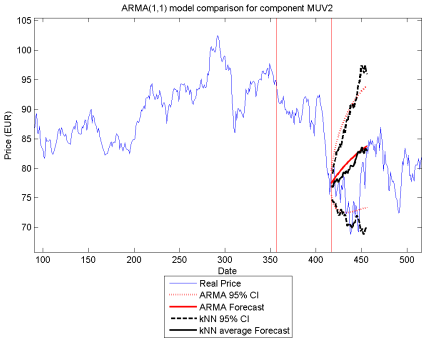
(C) 50% "history" MRK



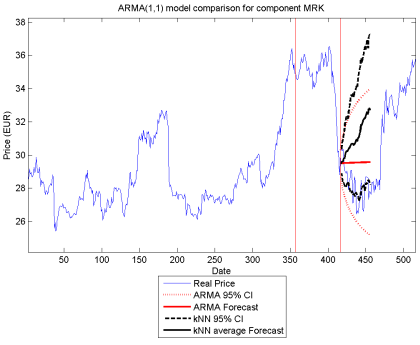
(D) 50% "history" MUV2



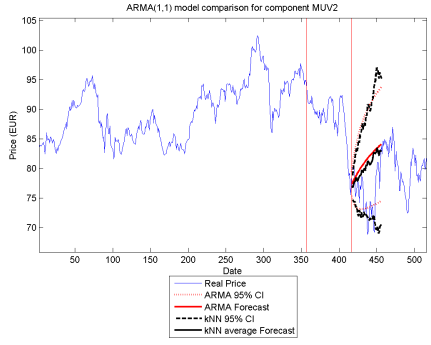
(E) 75% "history" MRK



(F) 75% "history" MUV2

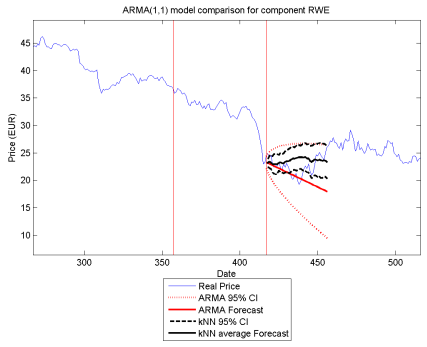


(G) 100% "history" MRK

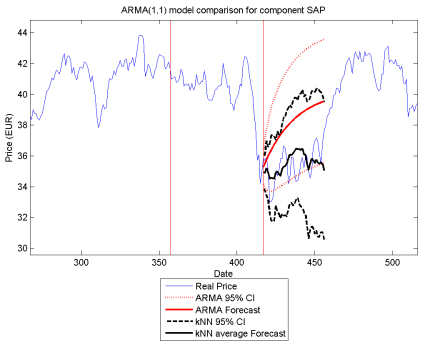


(H) 100% "history" MUV2

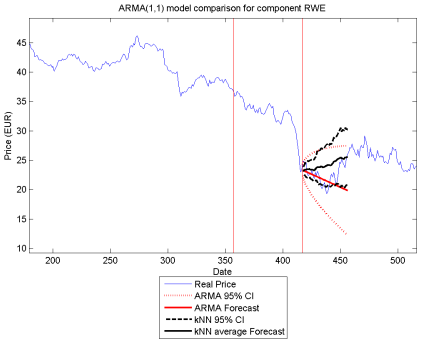
FIGURE C.11: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component MRK (all LEFT figures) and MUV2 (all RIGHT figures)



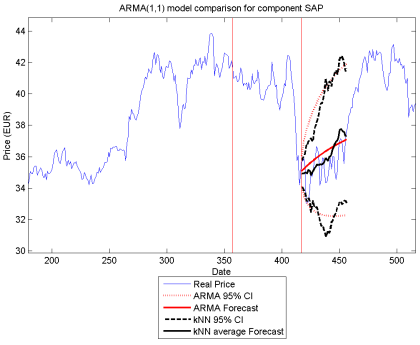
(A) 25% "history" RWE



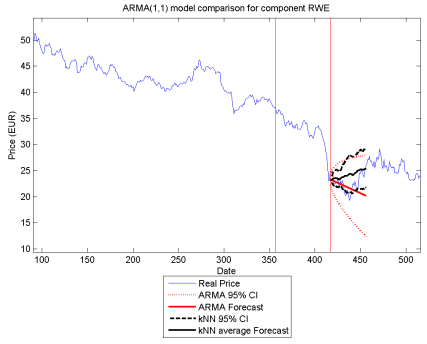
(B) 25% "history" SAP



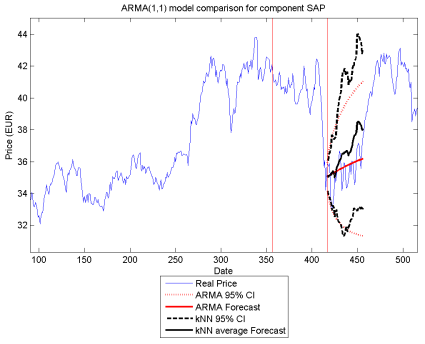
(C) 50% "history" RWE



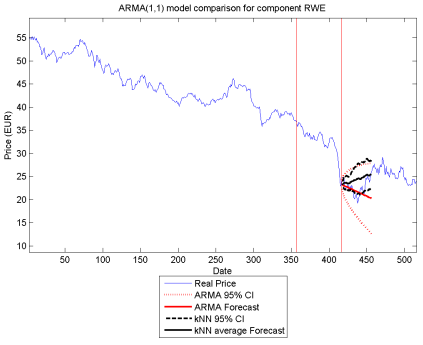
(D) 50% "history" SAP



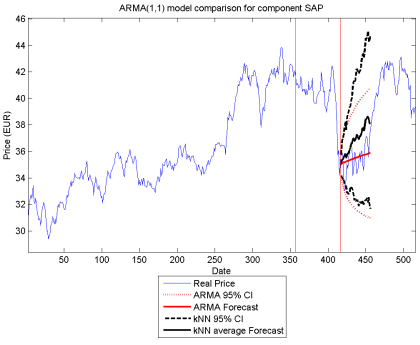
(E) 75% "history" RWE



(F) 75% "history" SAP

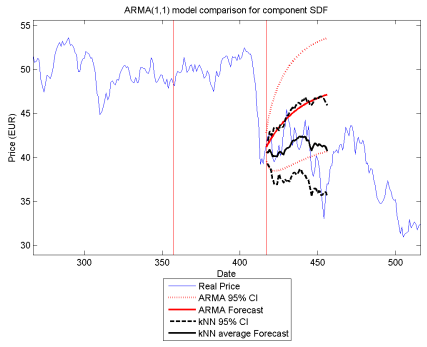


(G) 100% "history" RWE

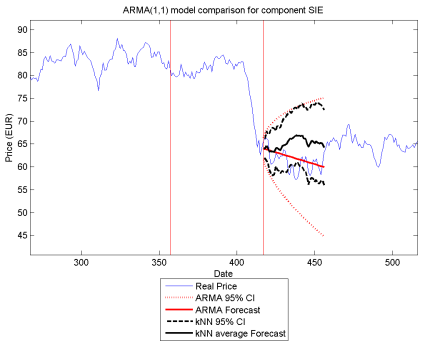


(H) 100% "history" SAP

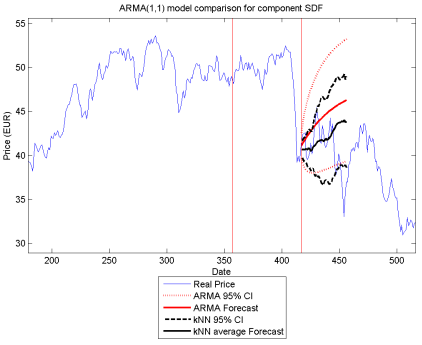
FIGURE C.12: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component RWE (all LEFT figures) and SAP (all RIGHT figures)



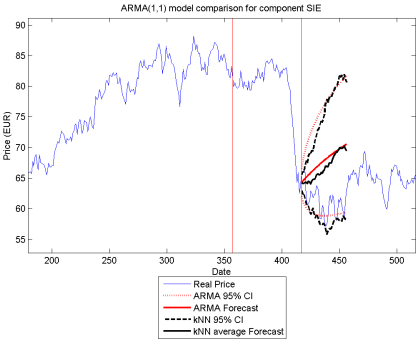
(A) 25% "history" SDF



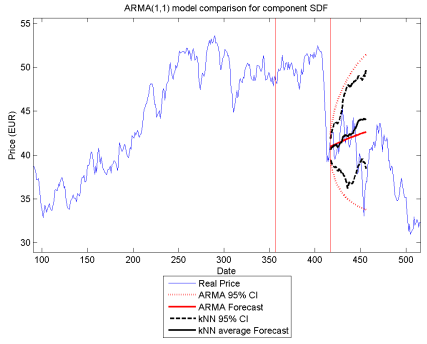
(B) 25% "history" SIE



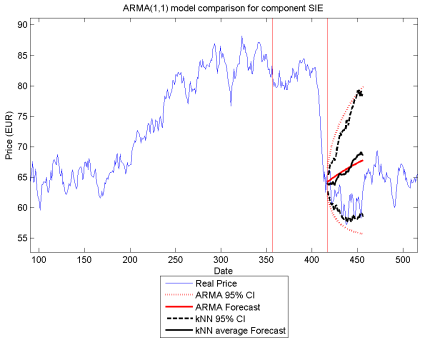
(C) 50% "history" SDF



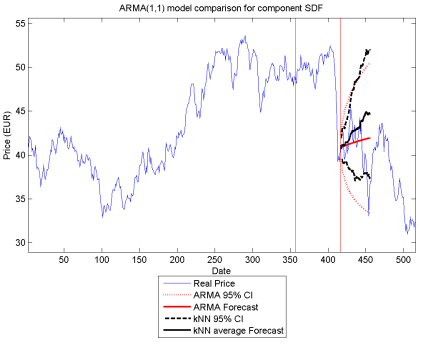
(D) 50% "history" SIE



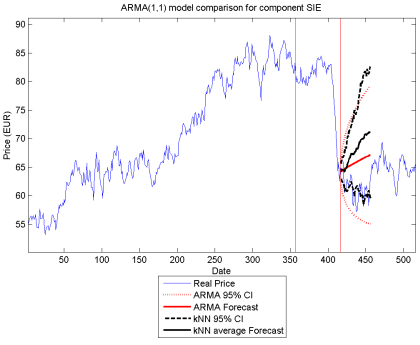
(E) 75% "history" SDF



(F) 75% "history" SIE

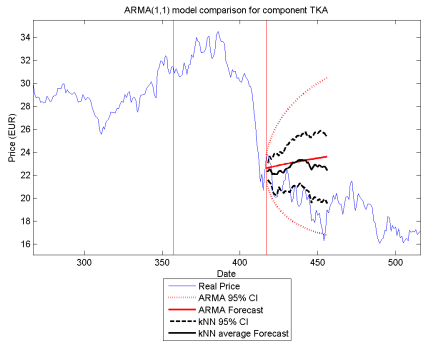


(G) 100% "history" SDF

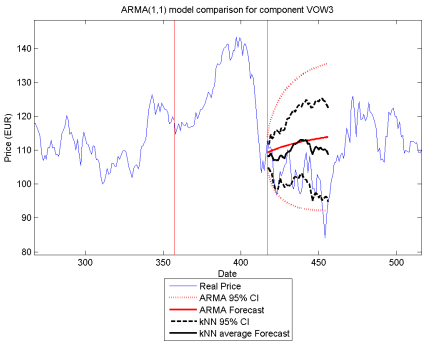


(H) 100% "history" SIE

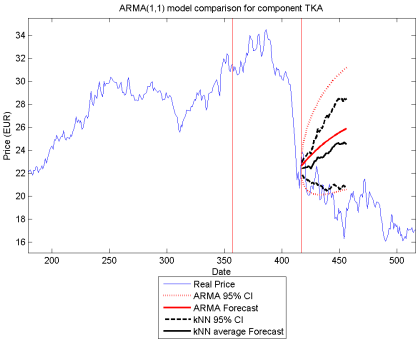
FIGURE C.13: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component SDF (all LEFT figures) and SIE (all RIGHT figures)



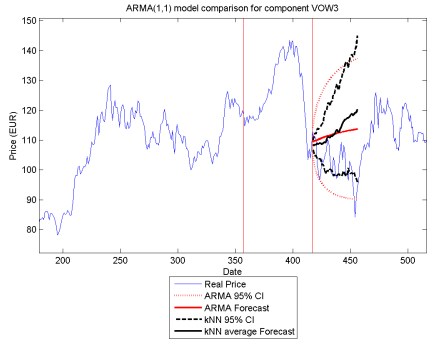
(A) 25% "history" TKA



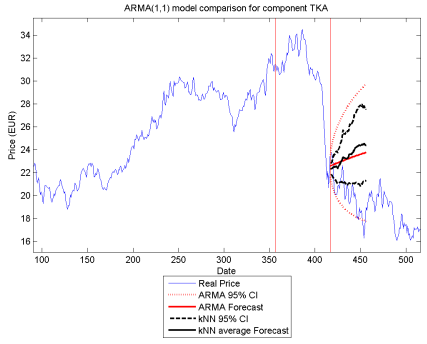
(B) 25% "history" VOW3



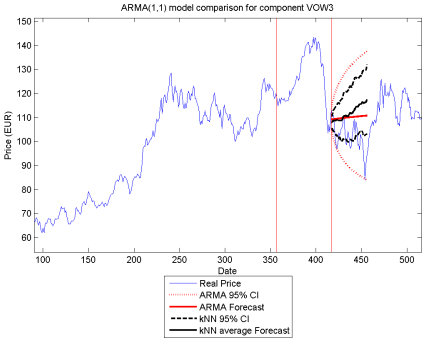
(C) 50% "history" TKA



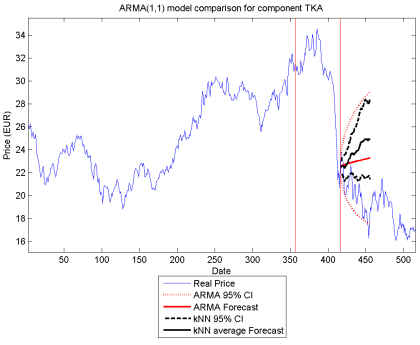
(D) 50% "history" VOW3



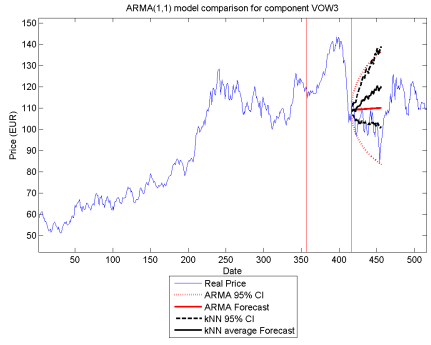
(E) 75% "history" TKA



(F) 75% "history" VOW3



(G) 100% "history" TKA

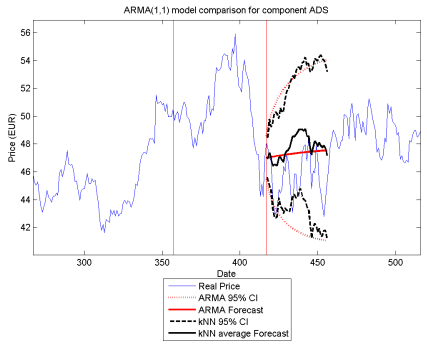


(H) 100% "history" VOW3

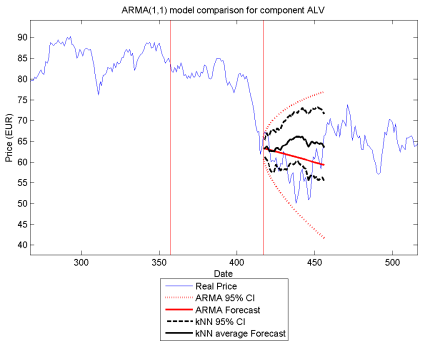
FIGURE C.14: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Correlation Distance) for component TKA (all LEFT figures) and VOW3 (all RIGHT figures)

Appendix D

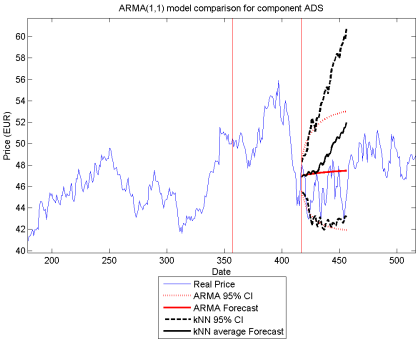
Graphs of comparison between the kNN approach and the ARMA(1,1) model (Cosine Similarity)



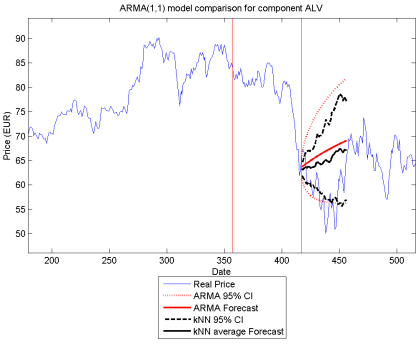
(A) 25% "history" ADS



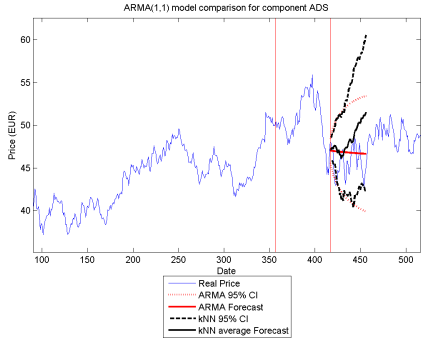
(B) 25% "history" ALV



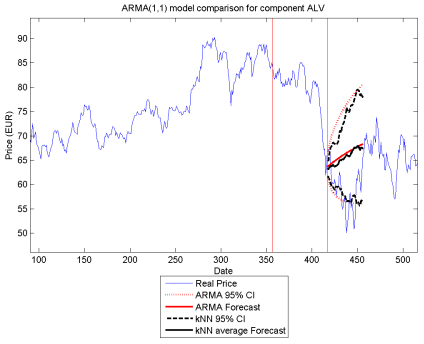
(C) 50% "history" ADS



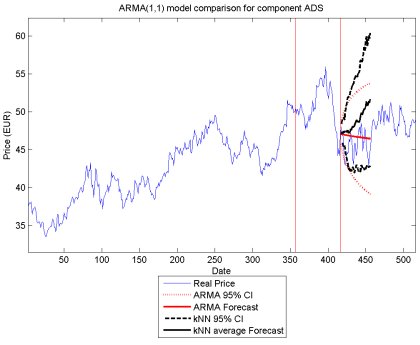
(D) 50% "history" ALV



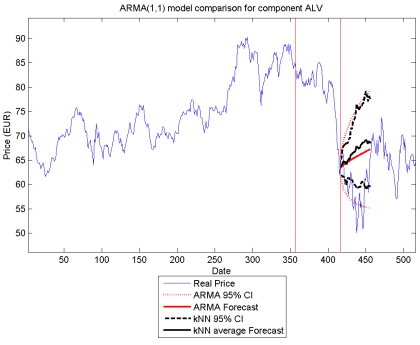
(E) 75% "history" ADS



(F) 75% "history" ALV

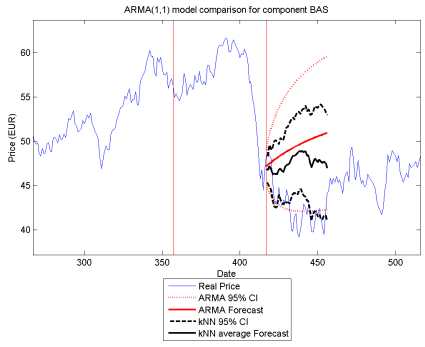


(G) 100% "history" ADS

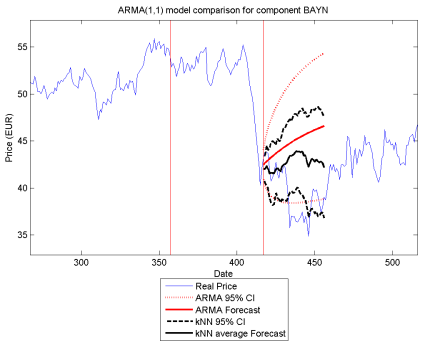


(H) 100% "history" ALV

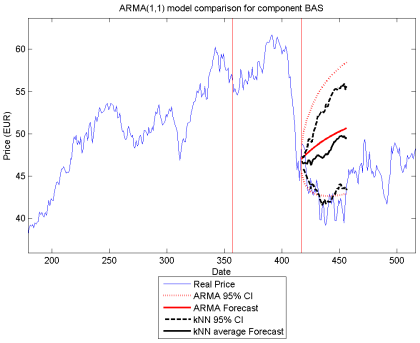
FIGURE D.1: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component ADS (all LEFT figures) and ALV (all RIGHT figures)



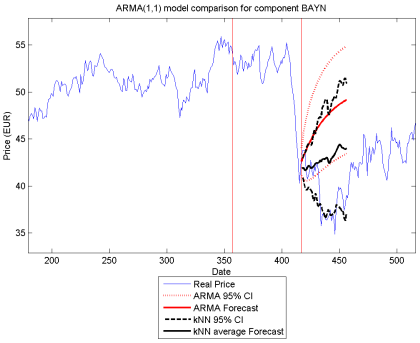
(A) 25% "history" BAS



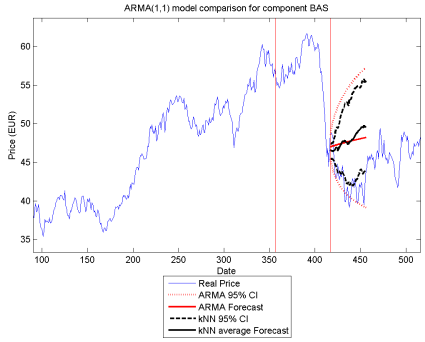
(B) 25% "history" BAYN



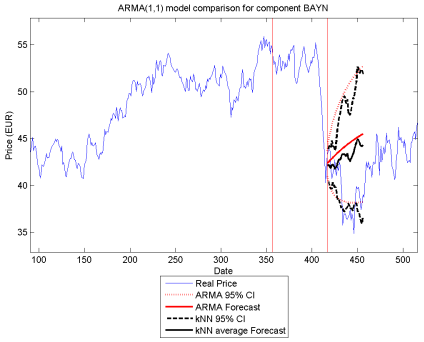
(C) 50% "history" BAS



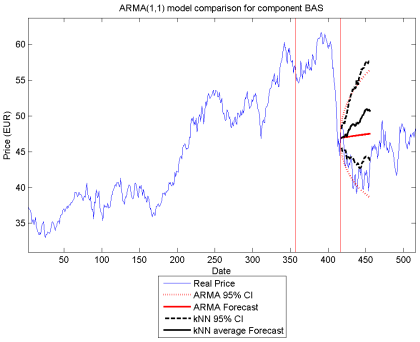
(D) 50% "history" BAYN



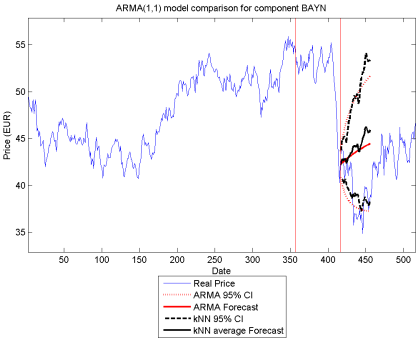
(E) 75% "history" BAS



(F) 75% "history" BAYN

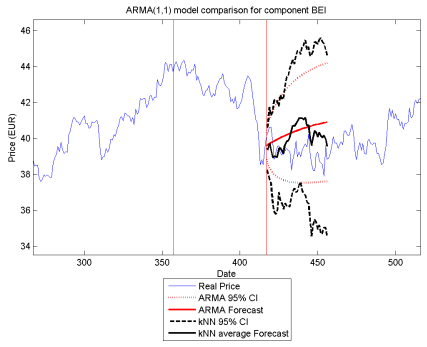


(G) 100% "history" BAS

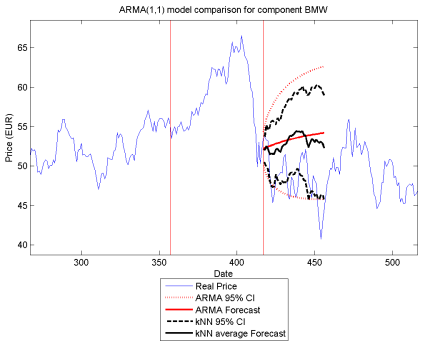


(H) 100% "history" BAYN

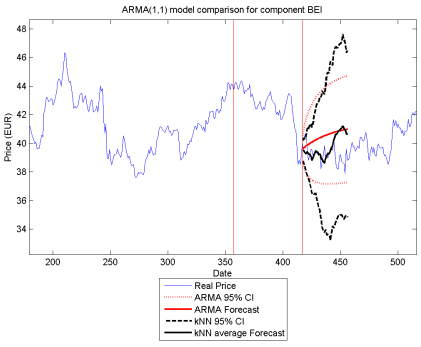
FIGURE D.2: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Euclidean Distance) for component BAS (all LEFT figures) and BAYN (all RIGHT figures)



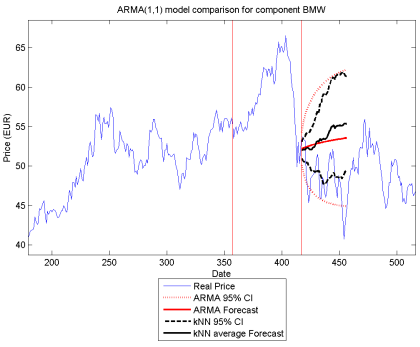
(A) 25% "history" BEI



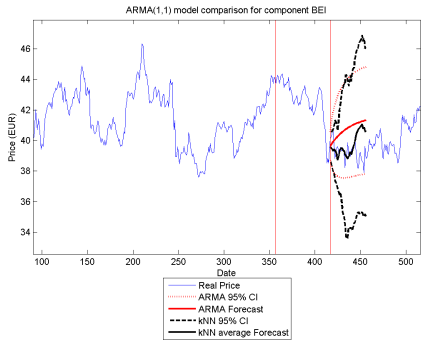
(B) 25% "history" BMW



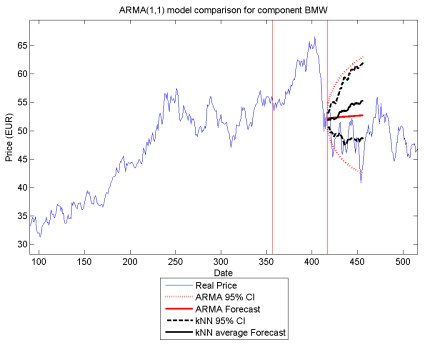
(C) 50% "history" BEI



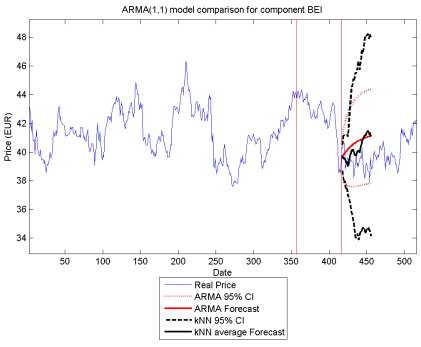
(D) 50% "history" BMW



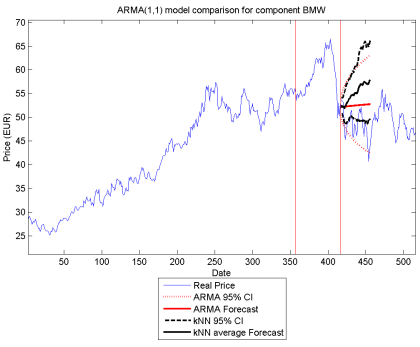
(E) 75% "history" BEI



(F) 75% "history" BMW

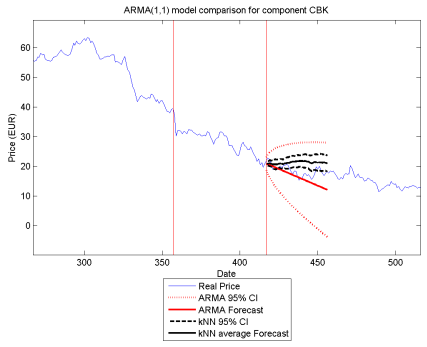


(G) 100% "history" BEI

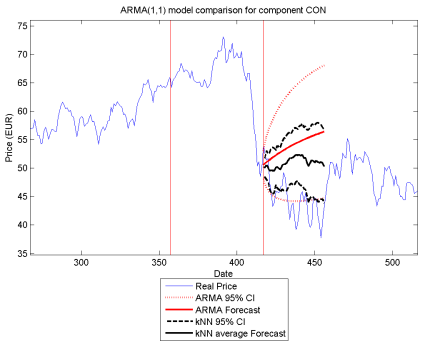


(H) 100% "history" BMW

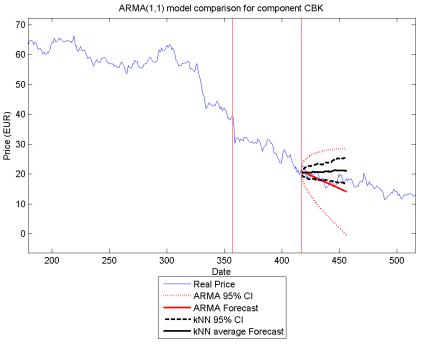
FIGURE D.3: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component BEI (all LEFT figures) and BMW (all RIGHT figures)



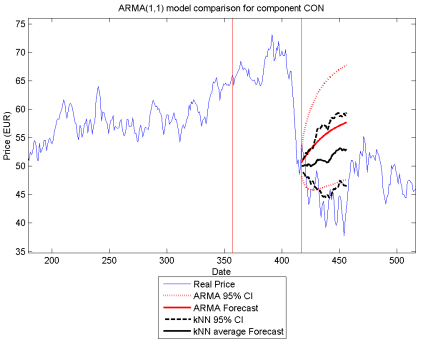
(A) 25% "history" CBK



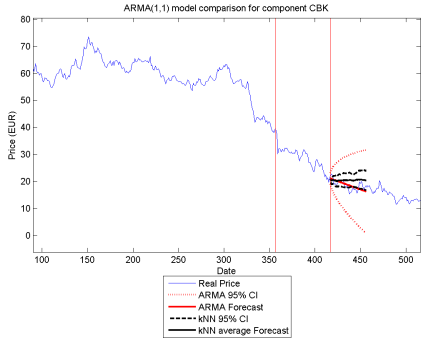
(B) 25% "history" CON



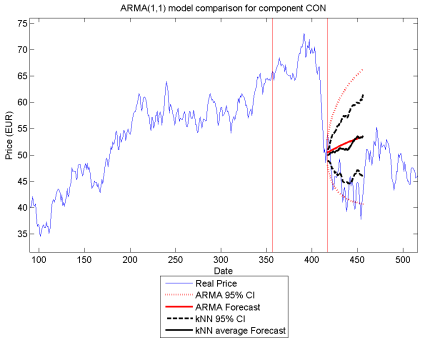
(C) 50% "history" CBK



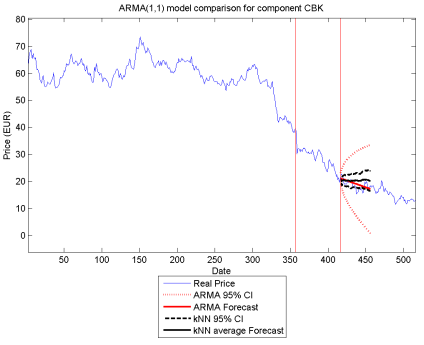
(D) 50% "history" CON



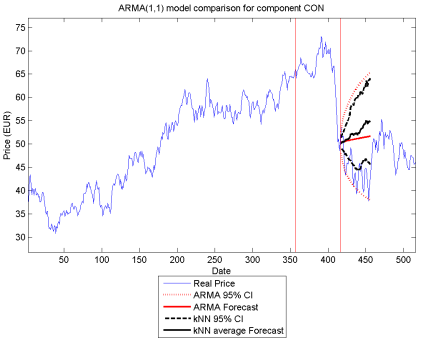
(E) 75% "history" CBK



(F) 75% "history" CON

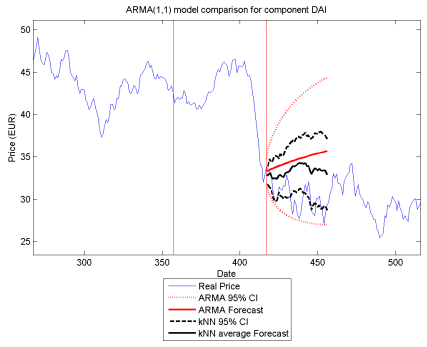


(G) 100% "history" CBK

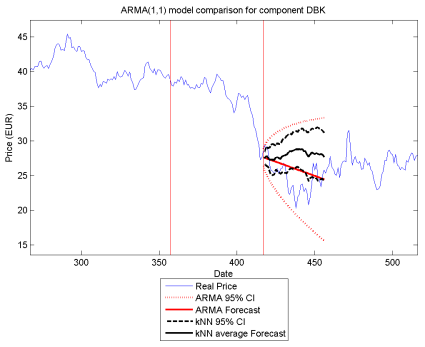


(H) 100% "history" CON

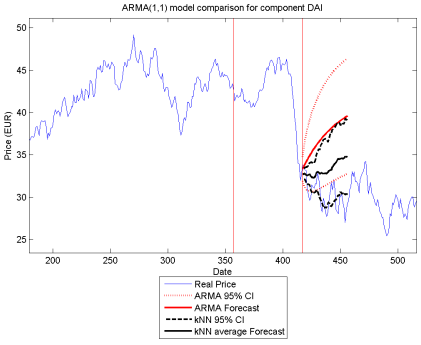
FIGURE D.4: 95% Confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component CBK (all LEFT figures) and CON (all RIGHT figures)



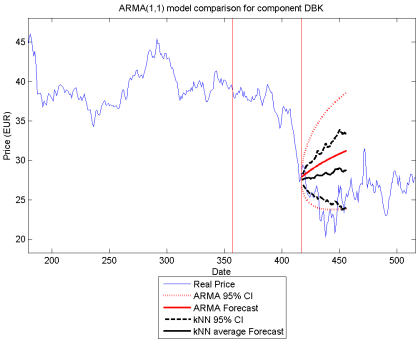
(A) 25% "history" DAI



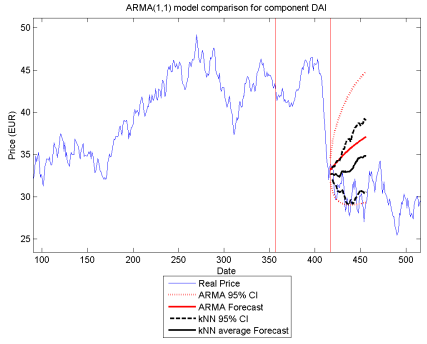
(B) 25% "history" DBK



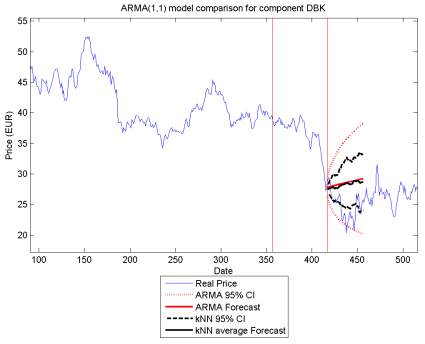
(C) 50% "history" DAI



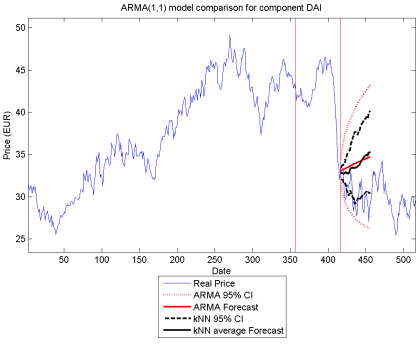
(D) 50% "history" DBK



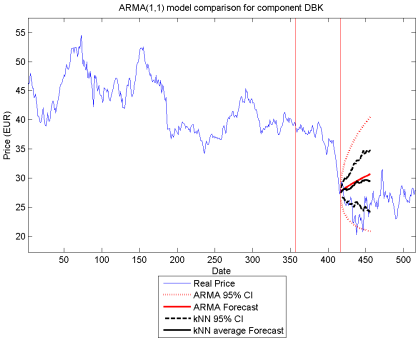
(E) 75% "history" DAI



(F) 75% "history" DBK

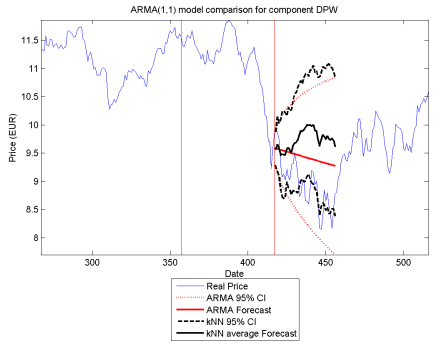


(G) 100% "history" DAI

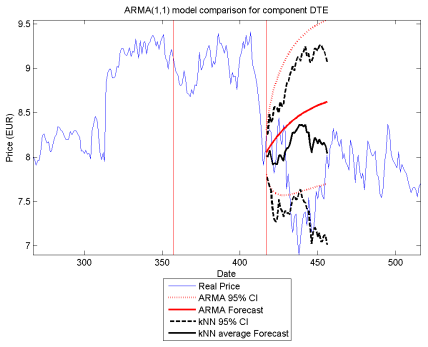


(H) 100% "history" DBK

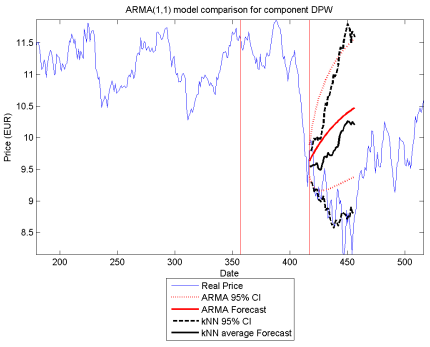
FIGURE D.5: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component DAI (all LEFT figures) and DBK (all RIGHT figures)



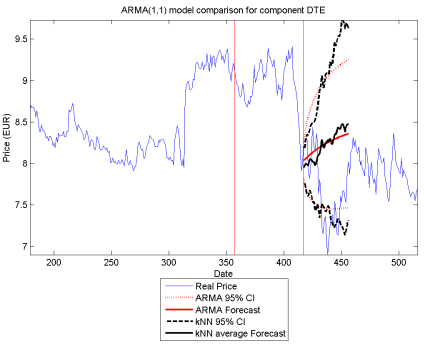
(A) 25% "history" DPW



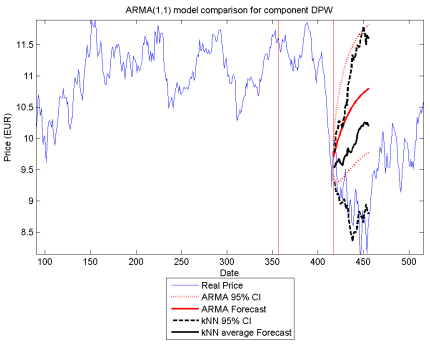
(B) 25% "history" DTE



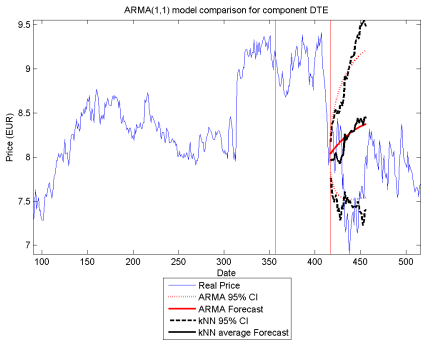
(C) 50% "history" DPW



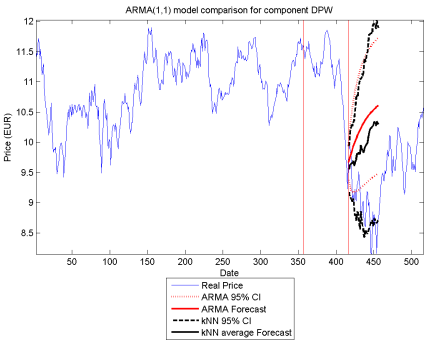
(D) 50% "history" DTE



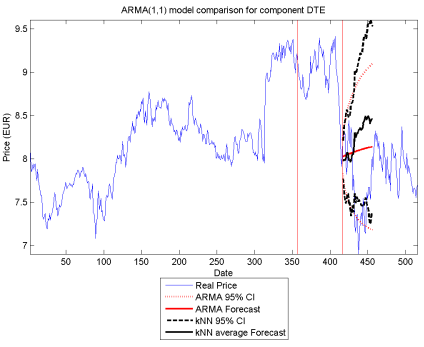
(E) 75% "history" DPW



(F) 75% "history" DTE

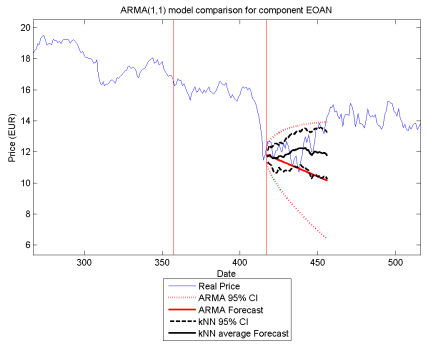


(G) 100% "history" DPW

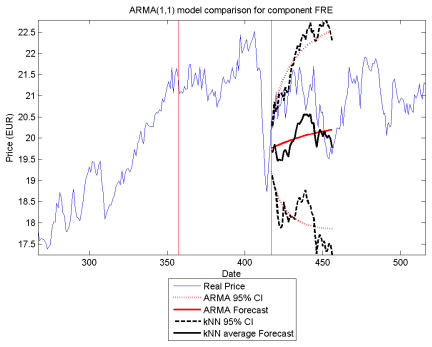


(H) 100% "history" DTE

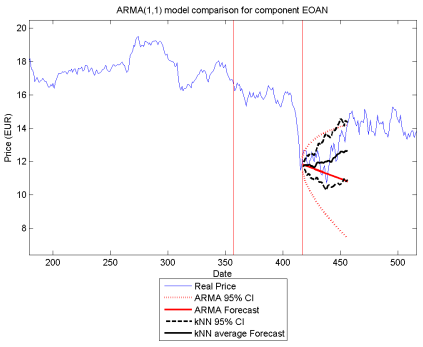
FIGURE D.6: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component DPW (all LEFT figures) and DTE (all RIGHT figures)



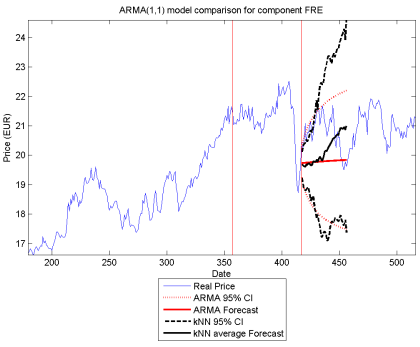
(A) 25% "history" EOAN



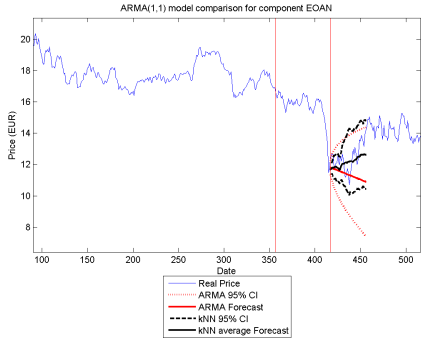
(B) 25% "history" FRE



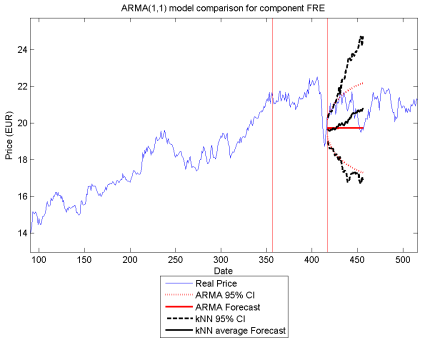
(C) 50% "history" EOAN



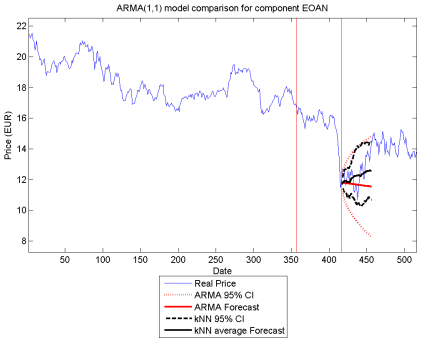
(D) 50% "history" FRE



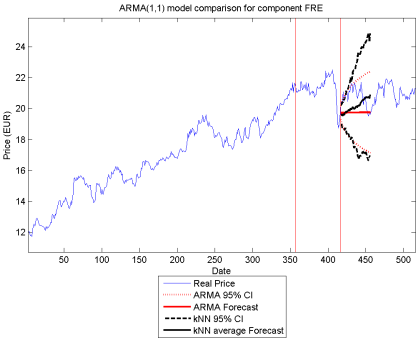
(E) 75% "history" EOAN



(F) 75% "history" FRE

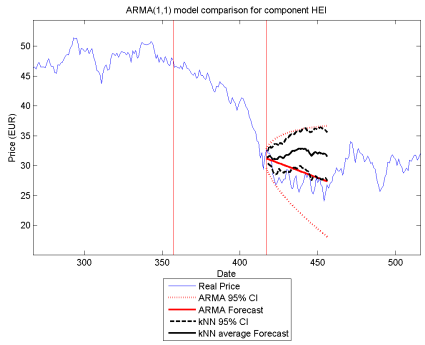


(G) 100% "history" EOAN

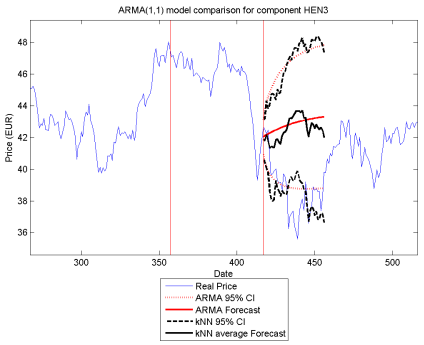


(H) 100% "history" FRE

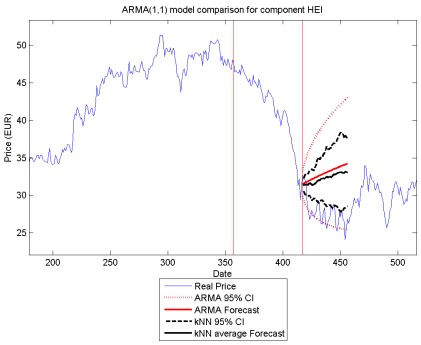
FIGURE D.7: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component EOAN (all LEFT figures) and FRE (all RIGHT figures)



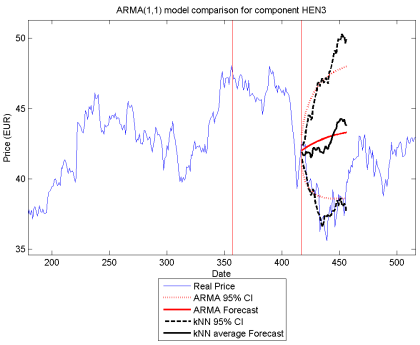
(A) 25% "history" HEI



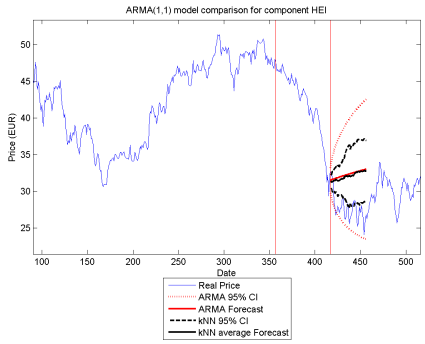
(B) 25% "history" HEN3



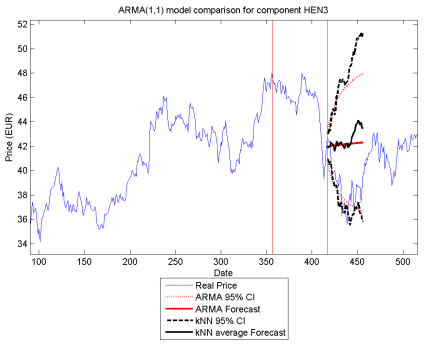
(C) 50% "history" HEI



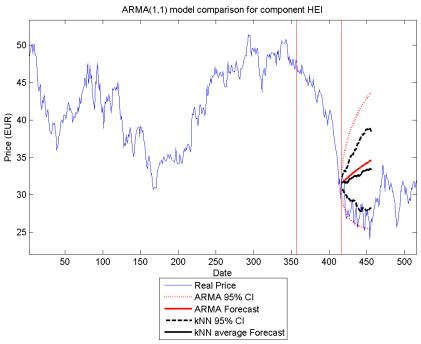
(D) 50% "history" HEN3



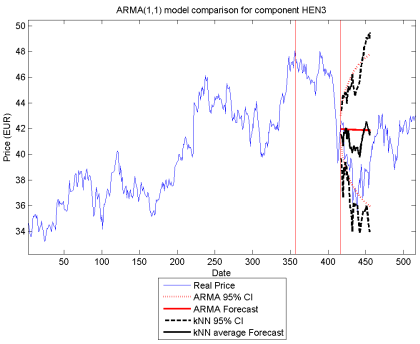
(E) 75% "history" HEI



(F) 75% "history" HEN3

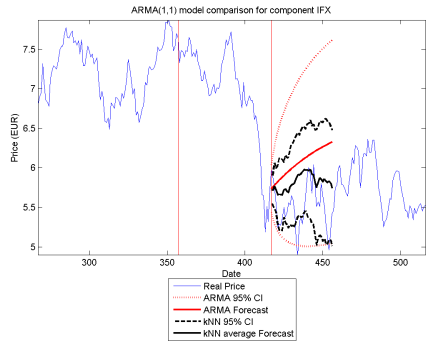


(G) 100% "history" HEI

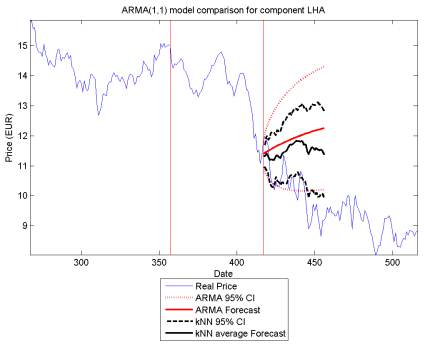


(H) 100% "history" HEN3

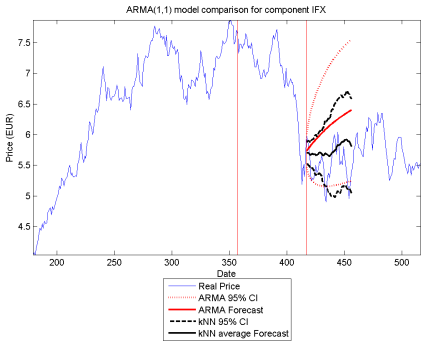
FIGURE D.8: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component HEI (all LEFT figures) and HEN3 (all RIGHT figures)



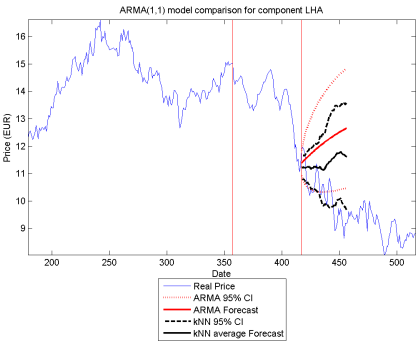
(A) 25% "history" IFX



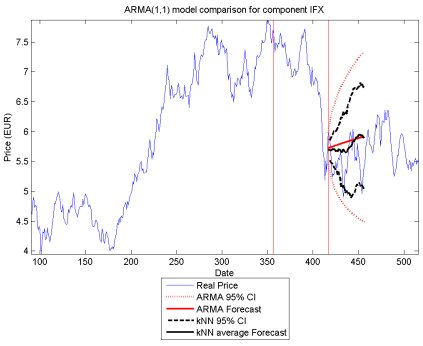
(B) 25% "history" LHA



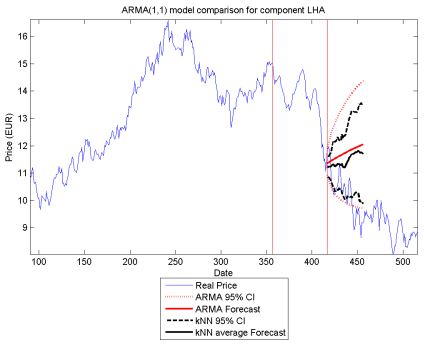
(C) 50% "history" IFX



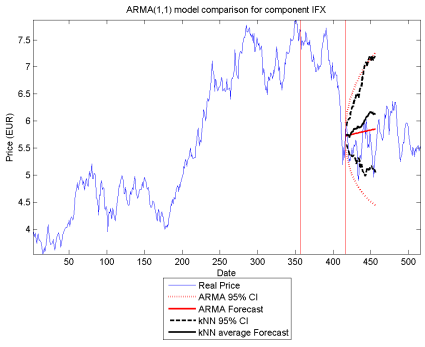
(D) 50% "history" LHA



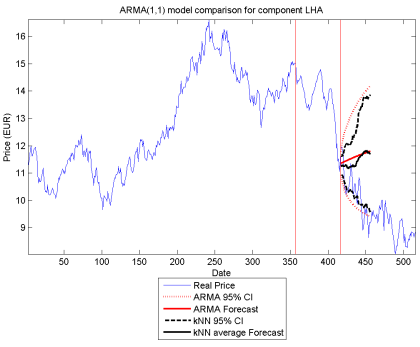
(E) 75% "history" IFX



(F) 75% "history" LHA

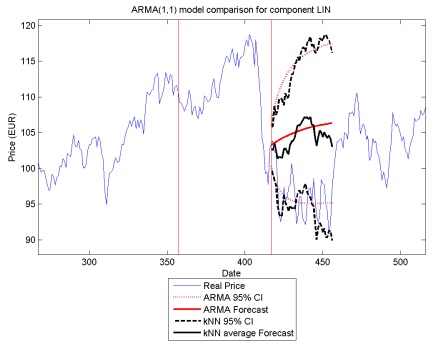


(G) 100% "history" IFX

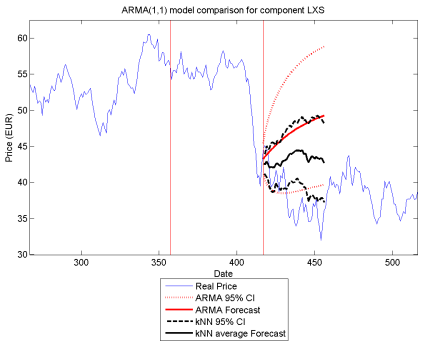


(H) 100% "history" LHA

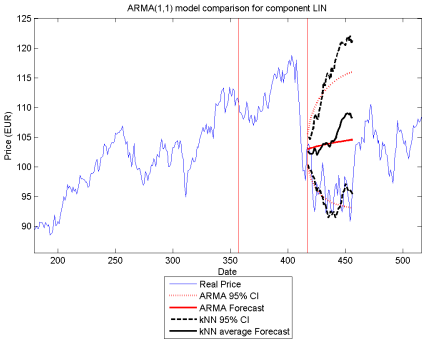
FIGURE D.9: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component IFX (all LEFT figures) and LHA (all RIGHT figures)



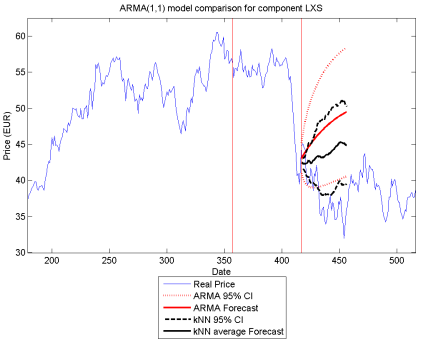
(A) 25% "history" LIN



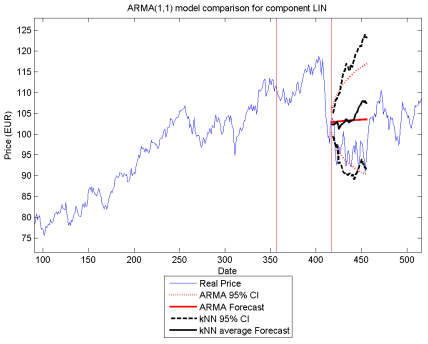
(B) 25% "history" LXS



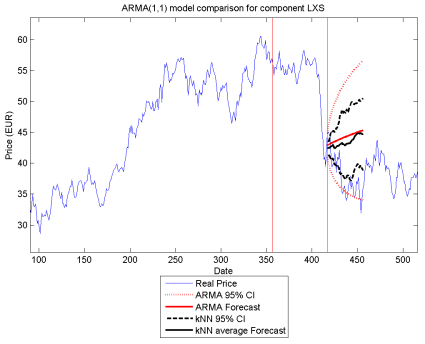
(C) 50% "history" LIN



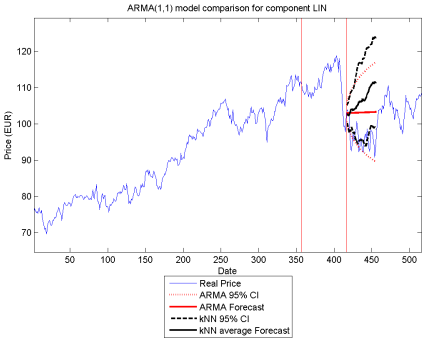
(D) 50% "history" LXS



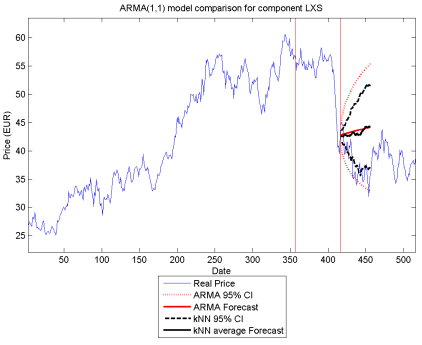
(E) 75% "history" LIN



(F) 75% "history" LXS

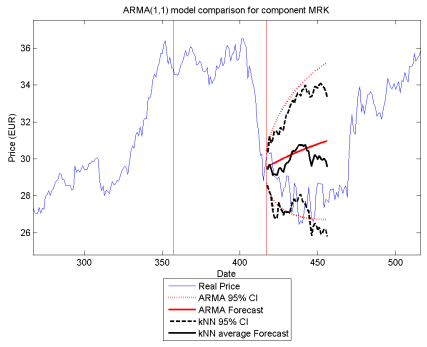


(G) 100% "history" LIN

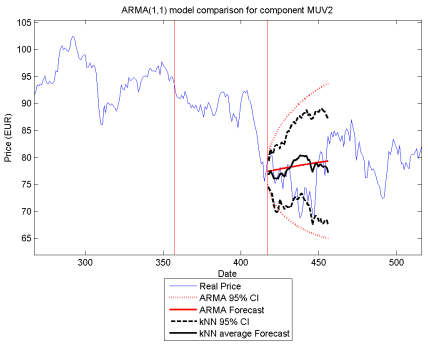


(H) 100% "history" LXS

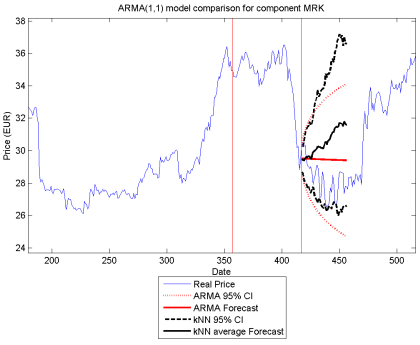
FIGURE D.10: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component LIN (all LEFT figures) and LXS (all RIGHT figures)



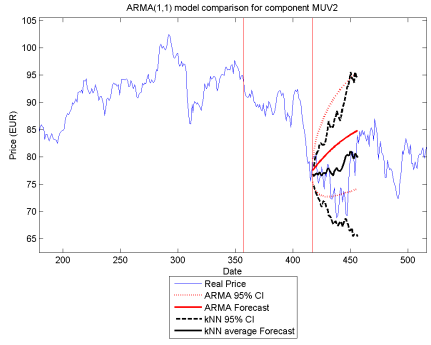
(A) 25% "history" MRK



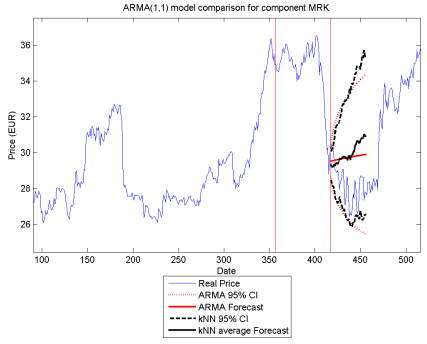
(B) 25% "history" MUV2



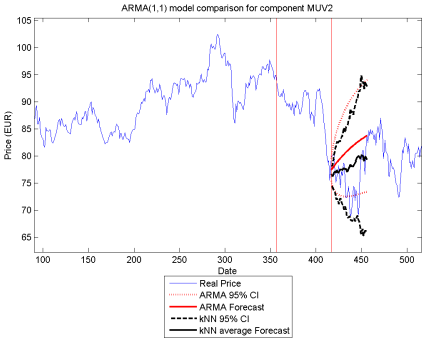
(C) 50% "history" MRK



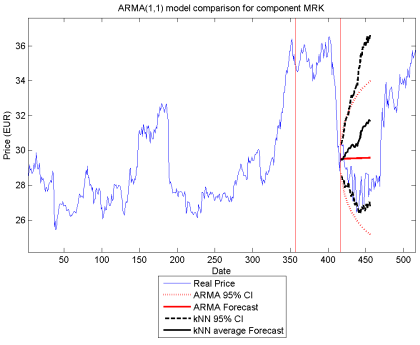
(D) 50% "history" MUV2



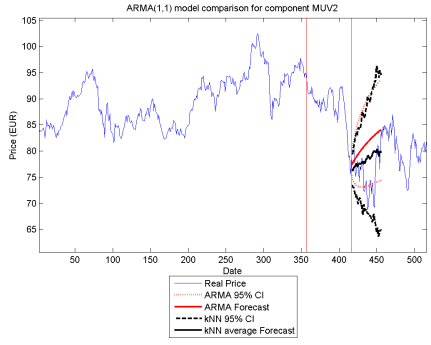
(E) 75% "history" MRK



(F) 75% "history" MUV2

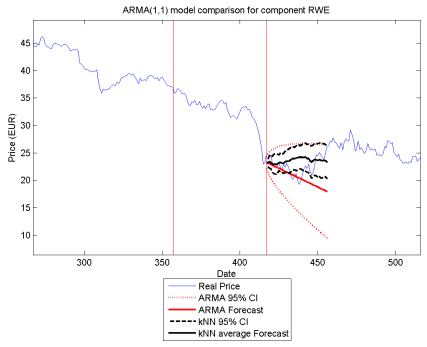


(G) 100% "history" MRK

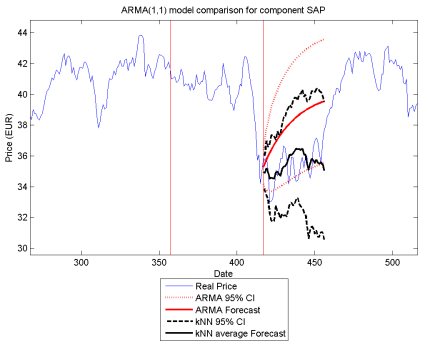


(H) 100% "history" MUV2

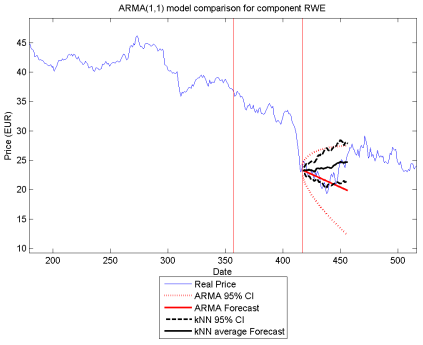
FIGURE D.11: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component MRK (all LEFT figures) and MUV2 (all RIGHT figures)



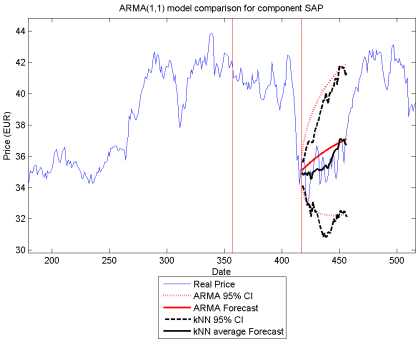
(A) 25% "history" RWE



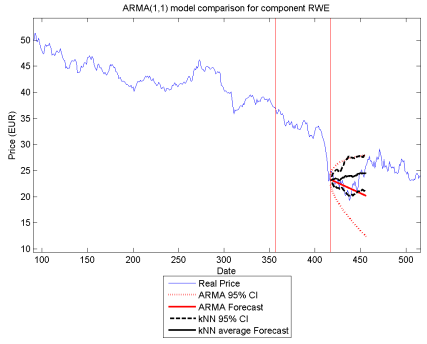
(B) 25% "history" SAP



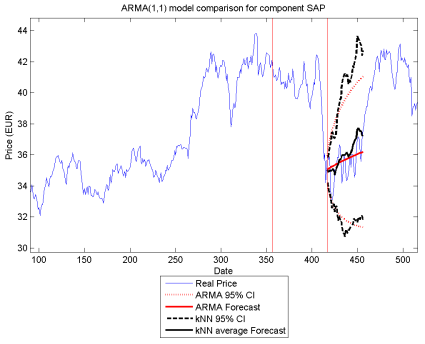
(C) 50% "history" RWE



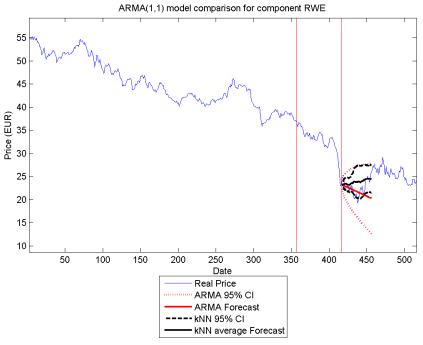
(D) 50% "history" SAP



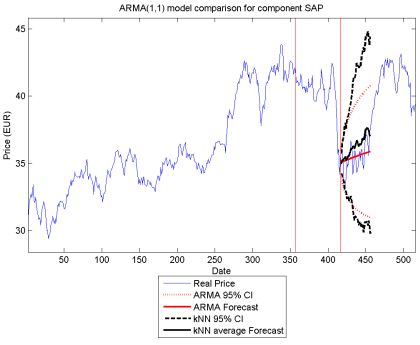
(E) 75% "history" RWE



(F) 75% "history" SAP

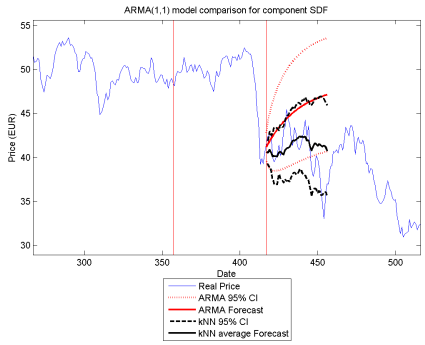


(G) 100% "history" RWE

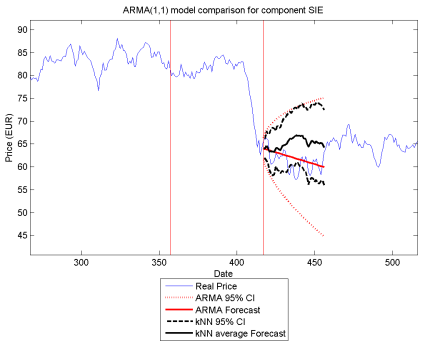


(H) 100% "history" SAP

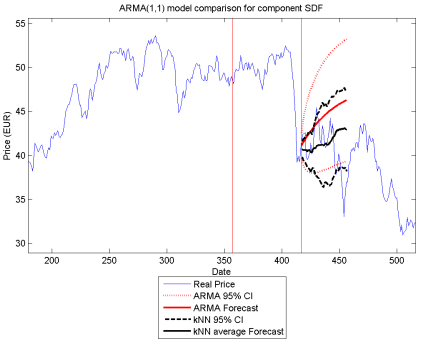
FIGURE D.12: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component RWE (all LEFT figures) and SAP (all RIGHT figures)



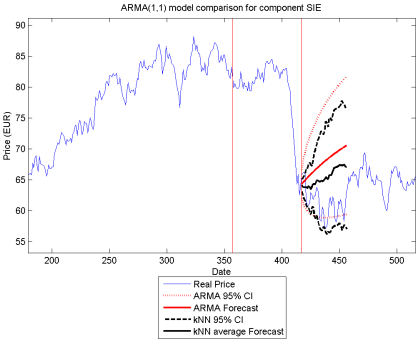
(A) 25% "history" SDF



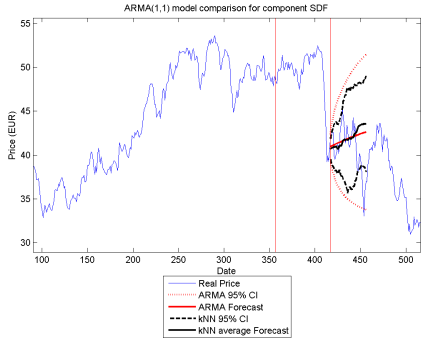
(B) 25% "history" SIE



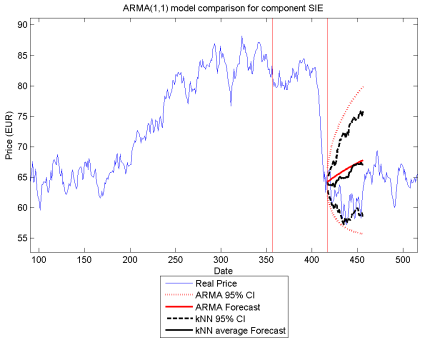
(C) 50% "history" SDF



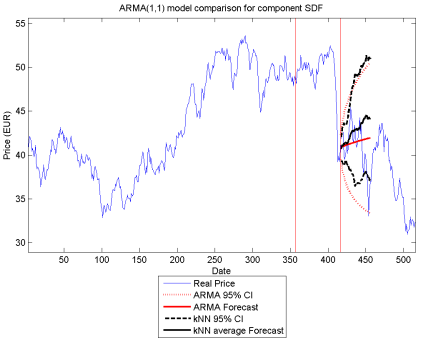
(D) 50% "history" SIE



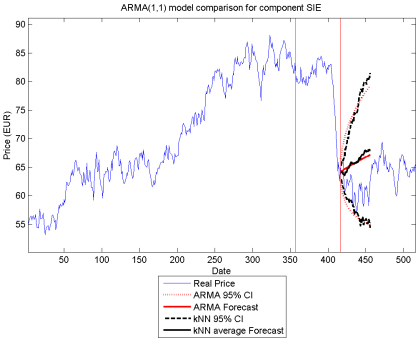
(E) 75% "history" SDF



(F) 75% "history" SIE

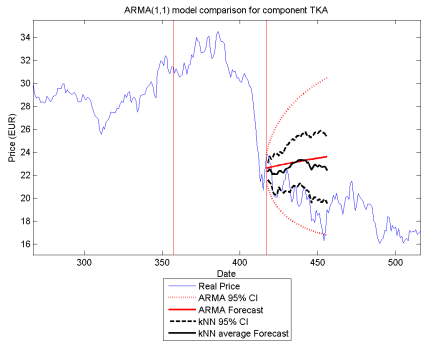


(G) 100% "history" SDF

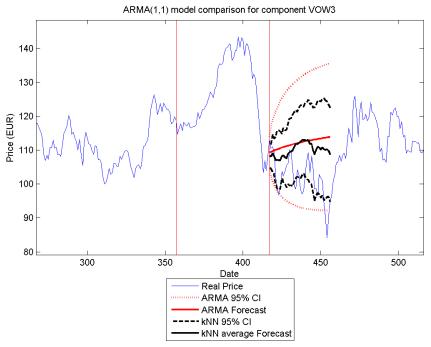


(H) 100% "history" SIE

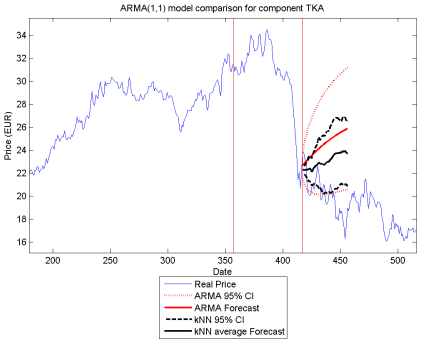
FIGURE D.13: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component SDF (all LEFT figures) and SIE (all RIGHT figures)



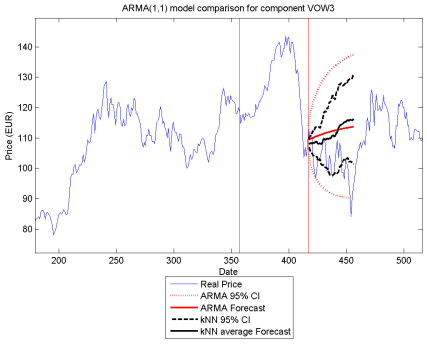
(A) 25% "history" TKA



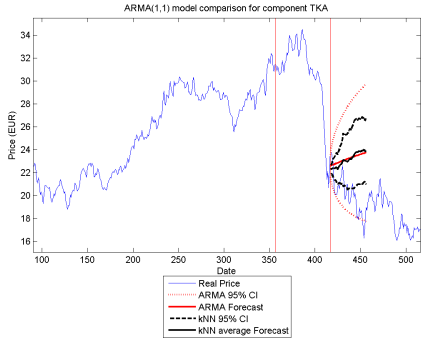
(B) 25% "history" VOW3



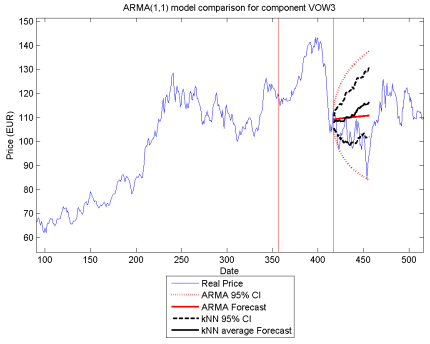
(C) 50% "history" TKA



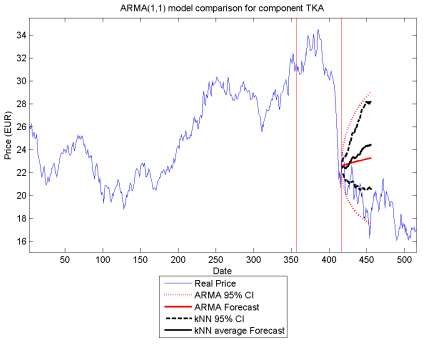
(D) 50% "history" VOW3



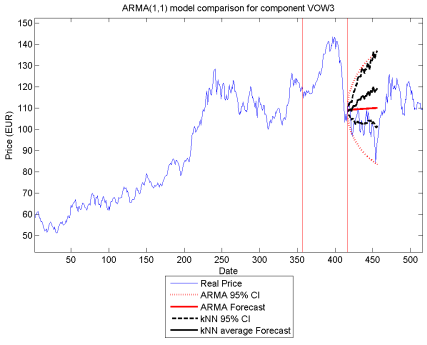
(E) 75% "history" TKA



(F) 75% "history" VOW3



(G) 100% "history" TKA



(H) 100% "history" VOW3

FIGURE D.14: 95% confidence interval comparison between ARMA(1,1) and kNN experiment (Cosine Similarity) for component TKA (all LEFT figures) and VOW3 (all RIGHT figures)

Bibliography

[1] Y. Shi, A. N. Gorban, and T. Y. Yang, "Is it possible to predict long-term success with k-nn? case study of four market indices (ftse100, dax, hangseng, nasdaq)," *Journal of Physics: Conference Series*, vol. 490, no. 1, p. 012082, 2014. [Online]. Available: http://stacks.iop.org/1742-6596/490/i=1/a=012082

[2] D. Enke and S. Thawornwong, "The use of data mining and neural networks for forecasting stock market returns," *Expert Systems with applications*, vol. 29, no. 4, pp. 927–940, 2005.

[3] H. White, "Economic prediction using neural networks: The case of ibm daily stock returns," in *Neural Networks, 1988., IEEE International Conference on*. IEEE, 1988, pp. 451–458.

[4] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.

[5] K. Alkhatib, H. Najadat, I. Hmeidi, and M. K. A. Shatnawi, "Stock price prediction using k-nearest neighbor (knn) algorithm," *International Journal of Business, Humanities and Technology*, vol. 3, no. 3, pp. 32–44, 2013.

[6] B. Qian and K. Rasheed, "Stock market prediction with multiple classifiers," *Applied Intelligence*, vol. 26, no. 1, pp. 25–33, 2007.

[7] N. Bhatia *et al.*, "Survey of nearest neighbor techniques," *arXiv preprint arXiv:1007.0085*, 2010.

[8] P. Hall, B. U. Park, and R. J. Samworth, "Choice of neighbor order in nearest-neighbor classification," *The Annals of Statistics*, pp. 2135–2152, 2008.

[9] E. Fix and J. L. Hodges Jr, "Discriminatory analysis–nonparametric discrimination: small sample performance," DTIC Document, Tech. Rep., 1952.

[10] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.

[11] T. M. Cover, "Rates of convergence for nearest neighbor procedures," in *Proceedings of the Hawaii International Conference on Systems Sciences*, 1968, pp. 413–415.

[12] ——, "Estimation by the nearest neighbor rule," *Information Theory, IEEE Transactions on*, vol. 14, no. 1, pp. 50–55, 1968.

[13] A. Djouadi and E. Bouktache, "A fast algorithm for the nearest-neighbor classifier," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 3, pp. 277–282, 1997.

[14] P. E. Hart, "The condensed nearest neighbor rule," *Information Theory, IEEE Transactions on*, vol. 14, no. 3, pp. 515–516, 1968.

[15] F. Angiulli, "Fast condensed nearest neighbor rule," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 25–32.

[16] E. Alpaydin, "Voting over multiple condensed nearest neighbors," *Artificial Intelligence Review*, vol. 11, no. 1-5, pp. 115–132, 1997.

[17] K. C. Gowda and G. Krishna, "The condensed nearest neighbor rule using the concept of mutual nearest neighborhood," *IEEE Transactions on Information Theory*, vol. 25, no. 4, pp. 488–490, 1979.

[18] G. Gates, "The reduced nearest neighbor rule," *Information Theory, IEEE Transactions on*, vol. 18, no. 3, pp. 431–433, May 1972.

[19] V. Pestov, "Is the k-nn classifier in high dimensions affected by the curse of dimensionality?" *Computers & Mathematics with Applications*, vol. 65, no. 10, pp. 1427–1437, 2013.

[20] P. Indyk, "Nearest neighbors in high-dimensional spaces," 2004.

[21] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 18, no. 6, pp. 607–616, 1996.

[22] Y. Liao and V. R. Vemuri, "Use of k-nearest neighbor classifier for intrusion detection," *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.

[23] G. Toker and O. Kirmemis, "Text categorization using k nearest neighbor classification," *Survey Paper, Middle East Technical University.*

[24] Y. Zhou, Y. Li, and S. Xia, "An improved knn text classification algorithm based on clustering," *Journal of computers*, vol. 4, no. 3, pp. 230–237, 2009.

[25] S. C. Bagui, S. Bagui, K. Pal, and N. R. Pal, "Breast cancer detection using rank nearest neighbor classification rules," *Pattern recognition*, vol. 36, no. 1, pp. 25–34, 2003.

[26] Y. Zeng, Y. Yang, and L. Zhao, "Pseudo nearest neighbor rule for pattern classification," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3587–3595, 2009.

[27] V. Vaidehi, S.Vasuhi, R.Kayalvizhi, K.Mariammal, Raghuraman.M.B, S. Raman.V, Meenakshi.L, and A. amd Thangamani.T, "Person authentication using face detection," in *Proceedings of the world congress on engineering and computer science.* International association of engineers, 2008, pp. 1166–1171.

[28] S. Xu and Y. Wu, "An algorithm for remote sensing image classification based on artificial immune b–cell network," *The international archives of the photogrammetry, remote sensing and spatial information sciences*, vol. 37, pp. 107–112, 2008.

[29] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[30] X. Geng, T. Liu, T. Qin, A. Arnold, H. Li, and H.-Y. Shum, "Query dependent ranking using k–nearest neighbor," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval.* ACM, 2008, pp. 115–122.

[31] K. Yang and C. Shahabi, "A pca-based similarity measure for multivariate time series," in *Proceedings of the 2nd ACM international workshop on Multimedia databases.* ACM, 2004, pp. 65–74.

[32] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, "Indexing multi-dimensional time-series with support for multiple distance measures," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2003, pp. 216–225.

[33] L. Wei and E. Keogh, "Semi-supervised time series classification," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 748–753.

[34] T. Liu, A. Moore, and A. Gray, "New algorithms for efficient high–dimensional nonparametric classification," *Journal of machine learning research*, vol. 7, pp. 1135–1158, 2006.

[35] Z. Li, K. Chan, and C. Wang, "Performance evaluation of the nearest feature line method in image classification and retrieval," *IEEE Transactions on pattern analysis & machine intelligence*, no. 11, pp. 1335–1349, 2000.

[36] W. Zheng, L. Zhao, and C. Zou, "Locally nearest neighbor classifiers for pattern classification," *Pattern recognition*, vol. 37, no. 6, pp. 1307–1309, 2004.

[37] J. McNames, "A fast nearest–neighbor algorithm based on a principal axis search tree," *Pattern analysis and machine intelligence, IEEE Transactions on*, vol. 23, no. 9, pp. 964–976, 2001.

[38] Y. Liaw, M. Leou, and C. Wu, "Fast exact k nearest neighbors search using an orthogonal search tree," *Pattern recognition*, vol. 43, no. 6, pp. 2351–2358, 2010.

[39] G. Das, K.-I. Lin, H. Mannila, G. Renganathan, and P. Smyth, "Rule discovery from time series." in *KDD*, vol. 98, 1998, pp. 16–22.

[40] J. L. Doob, *Stochastic processes*. New York Wiley, 1953, vol. 101.

[41] D. R. Cox and H. D. Miller, *The theory of stochastic processes*. CRC Press, 1977, vol. 134.

[42] D. Stirzaker, "Stochastic processes and models," *OUP Catalogue*, 2005.

[43] T. Hida, *Brownian motion*. Springer, 1980.

[44] S. Karlin, *A first course in stochastic processes*. Academic press, 2014.

[45] Z. Wu and N. E. Huang, "A study of the characteristics of white noise using the empirical mode decomposition method," in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 460, no. 2046. The Royal Society, 2004, pp. 1597–1611.

[46] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.

[47] B. Ryabko, J. Astola, and M. Malyutov, *Compression–based methods of statistical analysis and prediction of time series*. Tampere International Center for Signal Processing, 2010.

[48] B. Porat, *Digital processing of random signals: theory and methods*. Prentice-Hall, Inc., 1994.

[49] R. Durrett, *Probability: theory and examples*. Cambridge university press, 2010.

[50] C. R. Nelson and C. R. Plosser, "Trends and random walks in macroeconomic time series: some evidence and implications," *Journal of monetary economics*, vol. 10, no. 2, pp. 139–162, 1982.

[51] M. W. Watson, "Univariate detrending methods with stochastic trends," *Journal of monetary economics*, vol. 18, no. 1, pp. 49–75, 1986.

[52] Z. Wu, N. E. Huang, S. R. Long, and C.-K. Peng, "On the trend, detrending, and variability of nonlinear and nonstationary time series," *Proceedings of the National Academy of Sciences*, vol. 104, no. 38, pp. 14 889–14 894, 2007.

[53] E. R. Kanasewich, *Time sequence analysis in geophysics*. University of Alberta, 1981.

[54] A. S. Sedra and K. C. Smith, *Microelectronic circuits*. New York: Oxford University Press, 1998, vol. 1.

[55] D. O. Loftsgaarden, C. P. Quesenberry *et al.*, "A nonparametric estimate of a multivariate density function," *The Annals of Mathematical Statistics*, vol. 36, no. 3, pp. 1049–1051, 1965.

[56] L. David G. and W. Peter, "Working paper series on knn density estimation," August 2007, this research has been carried out within the Nccr Finrisk project on "Behavioural and Evolutionary Finance".

[57] W. H. Rogers, *Some convergence properties of k-nearest neighbor estimates*. Department of Statistics, Stanford University, 1978.

[58] E. F. Fama, L. Fisher, M. C. Jensen, and R. Roll, "The adjustment of stock prices to new information," *International economic review*, vol. 10, no. 1, pp. 1–21, 1969.

[59] M. Sewell, "History of the efficient market hypothesis," *RN*, vol. 11, no. 04, p. 04, 2011.

[60] B. Malkiel, "The efficient market hypothesis and its critics," *Journal of economic perspectives*, pp. 59–82, 2003.

[61] M. Borges, "Efficient market hypothesis in european stock markets," *The European journal of finance*, vol. 16, no. 7, pp. 711–726, 2010.

[62] M. Beechey, D. W. Gruen, and J. Vickery, *The efficient market hypothesis: a survey.* Reserve Bank of Australia, Economic Research Department, 2000.

[63] M. P. Taylor and H. Allen, "The use of technical analysis in the foreign exchange market," *Journal of international Money and Finance*, vol. 11, no. 3, pp. 304–314, 1992.

[64] S. H. Irwin and J. W. Uhrig, "Technical analysis-a search for the holy grail," in *Ncr-134 Conference on Applied Commodity Price Analysis, Forecasting, and Market Risk Management*, 1984.

[65] R. Curcio, C. Goodhart, D. Guilluame, and R. Payne, "Do technical trading rules generate profits," *Conclusions from the intra-day*, 1997.

[66] C.-H. Park and S. H. Irwin, "What do we know about the profitability of technical analysis?" *Journal of Economic Surveys*, vol. 21, no. 4, pp. 786–826, 2007.

[67] C. R. Harvey, "Predictable risk and returns in emerging markets," *Review of Financial studies*, vol. 8, no. 3, pp. 773–816, 1995.

[68] M. W. H.B. Xie, J.Z. Bian and H. Qiao, "Is technical analysis informative in uk stock market? evidence from decomposition-based vector autoregressive (dvar) model," *Journal of systems science and complexity*, vol. 27, no. 1, pp. 144–156, 2014.

[69] H. Bessembinder and K. Chan, "Market efficiency and the returns to technical analysis," *Financial management*, pp. 5–17, 1998.

[70] L. A. Teixeira and A. L. I. De Oliveira, "A method for automatic stock trading combining technical analysis and nearest neighbor classification," *Expert systems with applications*, vol. 37, no. 10, pp. 6885–6890, 2010.

[71] F. Longin and B. Solnik, "Is the correlation in international equity returns constant: 1960–1990?" *Journal of international money and finance*, vol. 14, no. 1, pp. 3–26, 1995.

[72] A. N. Gorban, E. V. Smirnova, and T. A. Tyukina, "Correlations, risk and crisis: From physiology to finance," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 16, pp. 3193–3217, 2010.

[73] R. S. Tsay, *Analysis of financial time series*. John Wiley & Sons, 2005, vol. 543.

[74] C. Dose and S. Cincotti, "Clustering of financial time series with application to index and enhanced index tracking portfolio," *Physica A: Statistical Mechanics and its Applications*, vol. 355, no. 1, pp. 145–151, 2005.

[75] J. G. Brida, N. Garrido, M. Deidda, and M. Pulina, "Exploring the dynamics of the efficiency in the italian hospitality sector. a regional case study," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9064–9071, 2012.

[76] F. Fernandez-Rodrıguez, S. Sosvilla-Rivero, and J. Andrada-Felix, "Exchange-rate forecasts with simultaneous nearest-neighbour methods: Evidence from the ems," *International Journal of Forecasting*, vol. 15, no. 4, pp. 383–392, 1999.

[77] F. Fernández-Rodríguez, S. Sosvilla-Rivero, and J. Andrada-Félix, *Nearest-neighbour predictions in foreign exchange markets*. Springer, 2004.

[78] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[79] M. Spiegel and L. Stephens, *Schaum's Outline of Statistics*, ser. Schaum's Outline Series. McGraw-Hill Education, 2007. [Online]. Available: https://books.google.co.uk/books?id=qdcBmgs3N3AC

[80] A. N. Gorban and A. Y. Zinovyev, "Elastic maps and nets for approximating principal manifolds and their application to microarray data visualization," in *Principal manifolds for data visualization and dimension reduction*. Springer, 2008, pp. 96–130.

[81] A. N. Gorban and A. Zinovyev, "Principal manifolds and graphs in practice: from molecular biology to dynamical systems," *International journal of neural systems*, vol. 20, no. 03, pp. 219–232, 2010.

[82] A. N. Gorban, A. Pitenko, and A. Zinovyev, "Vidaexpert: user-friendly tool for nonlinear visualization and analysis of multidimensional vectorial data," *arXiv preprint arXiv:1406.5550*, 2014.

[83] S. Kandel and R. F. Stambaugh, "On the predictability of stock returns: An asset-allocation perspective," National Bureau of Economic Research, Tech. Rep., 1995.

[84] A. Ang and G. Bekaert, "Stock return predictability: Is it there?" *Review of Financial studies*, vol. 20, no. 3, pp. 651–707, 2007.

[85] J. Wichard and M. Ogorzalek, "Time series prediction with ensemble models," in *Proceedings of International Joint Conference on Neural Networks (IJCNN 2004)*. Citeseer, 2004, pp. 1625–1629.

[86] J. McNames, "A nearest trajectory strategy for time series prediction," in *Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear Modeling*. KU Leuven Belgium, 1998, pp. 112–128.

[87] K. Judd and A. Mees, "On selecting models for nonlinear time series," *Physica D: Nonlinear Phenomena*, vol. 82, no. 4, pp. 426–444, 1995.

[88] F. Takens, *Detecting strange attractors in turbulence*. Springer, 1981.

[89] J. Stark, D. Broomhead, M. Davies, and J. Huke, "Takens embedding theorems for forced and stochastic systems," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 30, no. 8, pp. 5303–5314, 1997.

[90] H. L. Yap, "Takens' embedding theorem."

[91] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation," *Econometrica: Journal of the Econometric Society*, pp. 987–1007, 1982.

[92] T. Bollerslev, R. Y. Chou, and K. F. Kroner, "Arch modeling in finance: A review of the theory and empirical evidence," *Journal of econometrics*, vol. 52, no. 1-2, pp. 5–59, 1992.

[93] R. Engle, "Estimates of the variance of us inflation base on the arch model," *Universitiy of California, San Diego Discussion Paper*, pp. 80–114, 1980.

[94] P. Giot and S. Laurent, "Value-at-risk for long and short trading positions," *Journal of Applied Econometrics*, vol. 18, no. 6, pp. 641–663, 2003.

[95] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[96] D. B. Nelson, "Conditional heteroskedasticity in asset returns: A new approach," *Econometrica: Journal of the Econometric Society*, pp. 347–370, 1991.

[97] E. Ferenstein and M. Gasowski, "Modelling stock returns with ar-garch processes," *SORT. 2004, Vol. 28, Núm. 1 [January-June]*, 2004.

[98] Y. K. Tse, "Stock returns volatility in the tokyo stock exchange," *Japan and the World Economy*, vol. 3, no. 3, pp. 285–298, 1991.

[99] A. Antoniou and P. Holmes, "Futures trading, information and spot price volatility: evidence for the ftse-100 stock index futures contract using garch," *Journal of Banking & Finance*, vol. 19, no. 1, pp. 117–129, 1995.

[100] J.-C. Duan and H. Zhang, "Pricing hang seng index options around the asian financial crisis–a garch approach," *Journal of Banking & Finance*, vol. 25, no. 11, pp. 1989–2014, 2001.

[101] R. Ramasamy and D. Shanmugam Munisamy, "Predictive accuracy of garch, gjr and egarch models select exchange rates application," *Global Journal of Management And Business Research*, vol. 12, no. 15, 2012.

[102] P. Whitle, *Hypothesis testing in time series analysis*. Almqvist & Wiksells, 1951, vol. 4.

[103] G. E. Box and G. M. Jenkins, *Time series analysis: forecasting and control, revised ed.* Holden-Day, 1976.

[104] J. D. Hamilton, *Time series analysis*. Princeton university press Princeton, 1994, vol. 2.

[105] W. Enders, *Applied econometric time series*. John Wiley & Sons, 2008.

[106] W. H. Greene, *Econometric analysis*. Pearson Education India, 2003.

[107] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the atmospheric sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[108] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential equations, dynamical systems, and an introduction to chaos*. Academic press, 2012.