Expert Systems With Applications 82 (2017) 19–26

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Bias effect on predicting market trends with EMD

Dennis Carnelossi Furlaneto[a,*], Luiz S. Oliveira[a,*], David Menotti[a], George D.C. Cavalcanti[b]

[a] UFPR - Av. Cel. Francisco H. dos Santos, 210, Jardim das Americas, Curitiba-PR, 81531-970, Brazil
[b] UFPE - Av. Jornalista Anibal Fernandes, Cidade Universitria, Recife-PE, 50740-540,Brazil

ARTICLE INFO

Article history:
Received 27 December 2016
Revised 23 March 2017
Accepted 24 March 2017
Available online 25 March 2017

Keywords:
Finance
Time series
EEMD
Trend prediction
Machine learning

ABSTRACT

Financial time series are notoriously difficult to analyze and predict, given their non-stationary, highly oscillatory nature. In this study, we evaluate the effectiveness of the Ensemble Empirical Mode Decomposition (EEMD), the ensemble version of Empirical Mode Decomposition (EMD), at generating a representation for market indexes that improves trend prediction. Our results suggest that the promising results reported using EEMD on financial time series were obtained by inadvertently adding look-ahead bias to the testing protocol via pre-processing the entire series with EMD, which affects predictive results. In contrast to conclusions found in the literature, our results indicate that the application of EMD and EEMD with the objective of generating a better representation for financial time series is not sufficient to improve the accuracy or cumulative return obtained by the models used in this study.

1. Introduction

Financial markets are notoriously difficult to analyze and predict. Prediction on the non-stationary, highly oscillatory time series generated by financial instruments such as stocks and market indexes (Mikosch & Stărică, 2004) present one of the most popular (Sapankevych & Sankar, 2009) and important problems in time series research. This is still considered an open research problem, and one that if solved accurately, would have obvious real-world applications.

Prediction problems in these time series usually assume one of the two forms: trend classification or value regression. Historically, the number of studies focusing on regression problems outweighs the number of studies with the focus on trend classification (Kumar & Thenmozhi, 2006). This is counter-intuitive as systems with small regression errors (e.g. root mean squared error (RMSE) and mean absolute error (MAE)) could still lead to incorrect decision making. Small errors would be especially damaging when working with prediction on more mature markets, where the volatility and strength of stocks and market indexes movements are less pronounced than in emerging markets.

In an attempt to improve the accuracy and other metrics used to benchmark prediction models in financial markets, such as the *Trend Accuracy* and *Cumulative Return*, many researchers focused their efforts on benchmarking and selecting the best prediction models for the task, using a variety of different technical indicators in addition to the raw lagged values as independent variables (Hsu, Lessmann, Sung, Ma, & Johnson, 2016). These features are used for the purpose of increasing the accuracy of price trend classification and regression of future values (Kim, 2003). However, given the current state-of-the-art and the results obtained by researchers without and with the use of technical indicators, there is evidence against the informational value of financial technical indicators (Hsu et al., 2016).

The Ensemble Empirical Model Decomposition (EEMD) (Wu & Huang, 2009), the ensemble version of Empirical Model Decomposition (EMD) (Huang et al., 1998), applied with the goal of feature extraction, aims at creating features by extracting quasi-periodic components from signals. The components generated by this non-parametric method can be used as inputs to classification models, effectively removing most of the human bias from feature generation. This decomposition technique has been applied successfully in many fields and is especially useful for non-stationary series. A few studies have applied it to predictive tasks in the field of finance, reporting success in doing so.

The objective of this study is to evaluate the effectiveness of the EEMD at generating a different representation for financial time series used to improve movement prediction, more specifically for the market indexes in the Istanbul Market Index dataset (Akbilgic, Bozdogan, & Balaban, 2014). To do so, we designed and used a testing protocol that has no look-ahead bias, in contrast to other works in the literature. The *Accuracy* and the *Cumulative Return* obtained by the classifiers such as Linear SVM, RBF SVM, Random Forest,

* Corresponding authors.
E-mail addresses: dcfurlaneto@inf.ufpr.br, dennis.furlaneto@gmail.com (D. Carnelossi Furlaneto), lesoliveira@inf.ufpr.br (L.S. Oliveira), menotti@inf.ufpr.br (D. Menotti), gdcc@cin.ufpe.br (G.D.C. Cavalcanti).

http://dx.doi.org/10.1016/j.eswa.2017.03.053
0957-4174/

and Logistic Regression using the components from EEMD as input were reported and compared against the results obtained by the same models using lagged values.

The remainder of this article is structured as follows. Section 2 explains in details how EMD and its ensemble variant (EEMD) work. Section 3 is dedicated to explaining what is look-ahead bias, and how it can be inadvertently added to testing protocols. Sections 4 and 5 follow up with a brief literature review with applications of EMD in the industry, and an explanation of the dataset we used in our evaluations, respectively. Sections 6 and 7 present the experimental protocol we designed as well as the experiments we have carried out. Finally, Section 8 concludes the work.

2. Empirical Mode Decomposition and Ensemble Empirical Mode Decomposition

Empirical Mode Decomposition (EMD) is an adaptive method created to separate the spectrum of non-linear and non-stationary signals (Wu & Huang, 2009). It decomposes a given time series, or signal, in components with different frequencies and amplitudes, called Intrinsic Mode Functions (IMFs). IMFs have two properties that distinguish them from other signals:

- The number of extrema and zero crossings must differ at most by one.
- The mean value between the upper and lower envelope is zero.

These conditions make the IMFs quasi-periodic, similar to harmonic signals, with the biggest difference between them being that there is no guarantee that the IMFs will have the same amplitude and frequency along the time axis. These IMFs, or simply modes as they are also known for, are extracted from the original time series through a process called sifting, where the order of IMFs extraction is from high-frequency to low-frequency signals; as the component extraction process progress, the modes look more and more periodic and have less noise embedded in them. Algorithm 1 describes this process in details:

Algorithm 1 Empirical Mode Decomposition.

Require: $x(t)$
Ensure: IMFs
$IMFs = []$
$x(t)' = x(t)$
while x(t)' is not monotonic **do**
  Identify all the maxima and minima values of $x(t)'$
  Generate upper and lower envelopes, $e_{min}(t)$ and $e_{max}(t)$, with cubic spline interpolation.
  Compute point-by-point average of upper and lower envelopes: $m(t) = (e_{min}(t) + e_{max}(t))/2$
  Compute the difference between $x(t)'$ and $m(t)$: $h(t) = x(t)' - m(t)$
  if Stopping Criterion is reached **then**
    IMFs.append($h(t)$)
    $x(t) = x(t) - h(t)$
    $x(t)' = x(t)$
  else
    $x(t)' = h(t)$
  end if
end while
return IMFs

The Stopping Criterion for the extraction of each IMF consists of verifying whether or not the component h can be defined as an IMF, as well as optional criterion such as the component maintaining its characteristics after S additional number of siftings (S-number) and a maximum number of siftings. The sifting process runs iteratively, extracting IMFs from the signal until the residue becomes a monotonic function, a constant value or a function with only one extremum from which no more IMFs can be extracted (Huang et al., 2003).

EEMD operates very similarly to EMD, but instead of decomposing the original signal once, it decomposes various copies of the original signal with different white Gaussian noises added to it, and averages all the IMFs generated by decomposing each of those copies. The addition of the noise helps the sifting process to avoid mode mixing, which is one of the main problems of the conventional EMD technique.

Algorithm 2 describes succinctly how EEMD operates:

Algorithm 2 Ensemble Empirical Mode Decomposition.

Require: $x(t)$, N, $Noise\ strength$
IMFs = []
Copy $x(t)$ K times
Add white noise to the copies of $x(t)$
IMFs.append(EMD(composed signals))
return Mean(IMFs)

Fig. 1 shows the IMFs extracted from a portion of the S&P500 index time series that is used in this study. The IMFs were plotted from first to last component that is extracted from the series, where the last plot contains the residue.

In addition to the input signal, the result of the decomposition using EEMD is also affected by a few additional parameters (as show in Algorithm 2):

- **Ensemble size** (*K*): The number of replicas of the input signal to be used in the ensemble
- **Noise strength**: Standard deviation of the Gaussian random noise added to the original signal before the sifting process starts.
- **S-number (Stopping Criterion for EMD)**: For *S* consecutive iterations, the number of zero crossings and extrema differ at most by one, and these numbers stay the same.
- **Maximum number of siftings (Stopping Criterion for EMD)**: A maximum number of total iterations can be set. This is done to increase the speed of the algorithm, prevent oversifting and to prevent the sifting procedure from being in a never-ending loop.

The "S-number" and "Maximum number of siftings" parameters play an important role, affecting the number and form of the IMFs produced by the algorithm. No matter the combination of variables though, the upper bound for the total number of IMFs extracted from a signal will be close to $\log_2(nPoints)$ (Wu & Huang, 2009).

3. Look-ahead bias on financial time series analysis

Look-ahead bias can be defined as the inadvertent use of information that is not available until a later date; in other words, forecasting the future using future data. Look-ahead bias might be added to research protocols or backtests in subtle ways; as explained by Mahfoud and Mani (1996), commercially and publicly available financial data might contain look-ahead bias from the start, with data associated to Governmental economic indicators for example going through review processes that might modify past figures.

Aside from look-ahead bias added to the data itself, the use of certain techniques as a pre-processing step might also be problematic. As an example, normalization techniques are very popular pre-processing steps in studies with financial time series. The min-max and the z-score normalizations, arguably the two most popular normalization techniques, make use of statistical variables

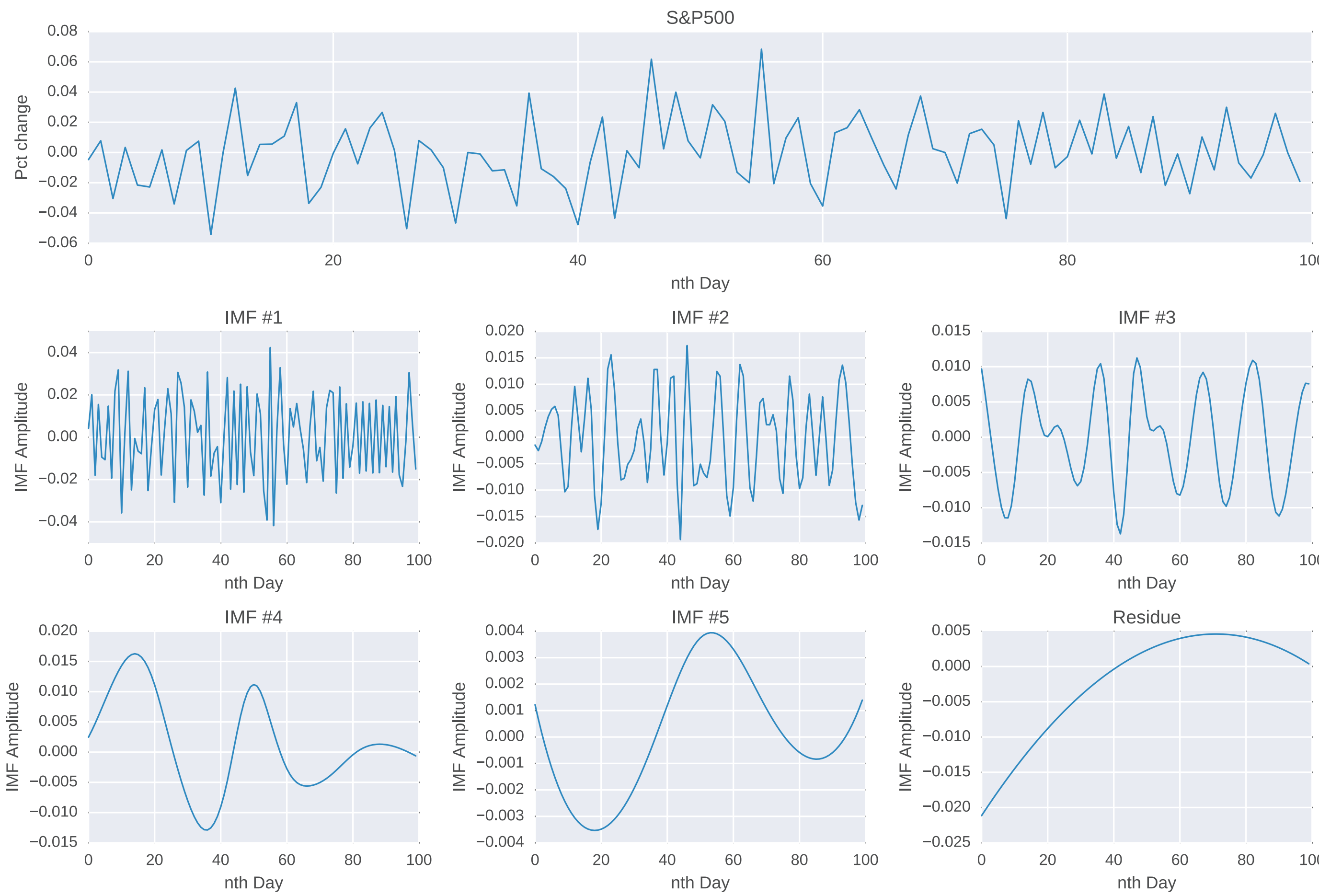

Fig. 1. EEMD decomposition applied to 100 days of data from the S&P500 index.

that might change as new information is added to time series, such as the minimum and maximum values, as well as the standard deviation. Normalizing the entire time series will add look-ahead bias to the evaluating protocol as variables that might change over time (min, max and standard deviation) are known and fixed from the start. For the min-max normalization in specific, note that using the percentage change, or Rate of Change (RoC), of the prices instead of the prices themselves as input minimizes the effect of the look-ahead bias as the minimum and maximum values change much less overtime when compared to the prices, since trend component is eliminated from the time series.

The general case is that extra care must be taken when using techniques that makes use of information only available at time t_1 to modify data points at time t_0 (where $t_1 > t_0$). Another example of such method is Empirical Mode Decomposition (EMD), which adds look-ahead bias if used inappropriately. As part of its algorithm, EMD performs successive searches for local minima and maxima, with a subsequent spline interpolation between those points to generate an upper and lower envelopes of the signal. Because EMD stores and subtracts the highest-frequency signal from the original signal on each iteration to create the IMFs, the higher-order components (lower frequency) will be generated by interpolating points that are far away from each other in the original series. Through these successive interpolations, future information is embedded on the IMFs.

Note that the existence of look-ahead bias might or might not affect the results obtained using a particular protocol, but the fact that its existence might heavily skew results should be enough for striving to remove it. For this reason, it is crucial that experimental protocols using EMD or its ensemble variation EEMD, and other pre-processing techniques for that matter, be carefully crafted to account for algorithmic peculiarities, in such a way that bias is not accidentally added.

Fig. 2 depicts from a high level perspective the protocol used in the literature (Al-Hnaity & Abbod, 2015; Fenghua, Jihong, Zhifang, & Xu, 2014; Xiong, Bao, Hu, Zhang, & Zhang, 2011; Yu, Wang, & Lai, 2008):

In Fig. 2, normalization methods might or might not be used as part of the protocol, and thus it was represented by a fading box. The dataset split and the way the models are trained and tested vary as well. However, using EMD as a pre-processing step seems to be common practice.

The addition of bias of such nature is unfortunately often overlooked in research works found in the literature, and protocols that contain look-ahead bias due to the usage of EMD seems to be prevalent.

4. Brief literature review on non-parametric decomposition techniques

Signal Decomposition techniques are used to deconstruct and represent signals as various components, each with different characteristics and associated with the underlying cyclical nature of the original signal. These methods can be used for a variety of tasks, from denoising the signal to making inferences about its periodic behavior and predictive tasks from the extraction of fetal heart signals from maternal ECG (Ghodsi, Hassani, & Sanei, 2010), analysis of seismic signals (Wang, Zhang, Yu, & Zhang, 2012) to prediction

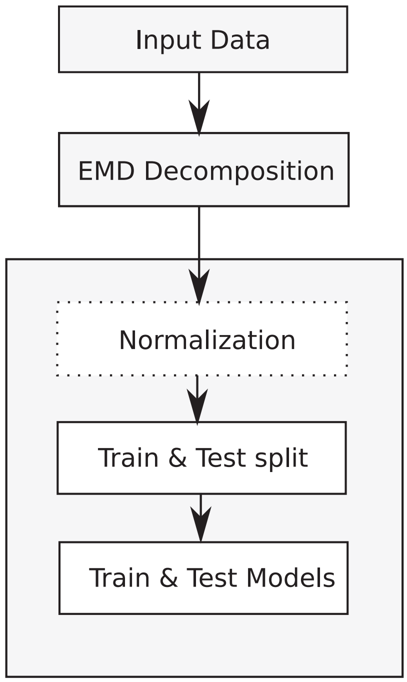

Fig. 2. Protocol with look-ahead bias.

on financial time series (Fenghua et al., 2014) being just a few applications.

Decomposition techniques can be divided into two main categories: Parametric and Non-parametric. Parametric methods make initial assumptions about the characteristics of the decomposed components, such as modeling these signals as sinusoidal waves of different amplitudes and frequencies (e.g. Fourier Transform). Parametric techniques are over-represented in the works found in the literature, where Fourier Transform and Wavelets are the most popular representatives.

Non-parametric techniques, on the other hand, make no a priori assumption for the generated components but are computationally more intensive than parametric models. These models are particularly good at decomposing non-linear, non-stationary time series due to their higher flexibility when compared to parametric models. The most well-known non-parametric techniques are the EEMD and Singular Spectrum Analysis (SSA) (Vautard, Yiou, & Ghil, 1992), with the EEMD being the most under-represented out of these two techniques. The main reason for this under-representation can be attributed to EEMD's very recent history: it was developed in 2009 (Wu & Huang, 2009) as an improvement to Empirical Mode Decomposition (EMD) was introduced in 1998 (Huang et al., 1998), while SSA was proposed in 1992.

Works in different domains have reported promising results with the use of non-parametric decomposition techniques. EEMD and SSA have been used for Stock Price Prediction (Fenghua et al., 2014), where the authors reported a trend prediction accuracy of approximately 68% with a combination of SSA and SVM and 63% using a combination of EEMD and SVM, both superior to accuracy obtained by SVM with raw data. In their review work (Lei, Lin, He, & Zuo, 2013), the authors explain that EMD has been widely applied and studied in fault diagnosis of rotating machinery. Tang, Wang, and Yu (2011) have used EEMD and Least Squares Support Vector Regression (LSSVR) to predict Nuclear Energy Consumption, with results reporting a reduction of 40% in the RMSE in comparison to LSSVR using raw data. These results show how these techniques can be used effectively in the analysis and prediction of time series, specially for noisy and non-stationary signals.

Table 1
Istanbul stock exchange dataset.

Market Index	Description
ISE100_TL	Istanbul Stock Market Index(TL)
ISE100_USD	Istanbul Stock Market Index(USD)
S&P 500	Standard & Poor's 500
DAX	German Stock Market Index
FTSE	London's Stock Market Index
NIKKEI	Tokyo's Stock Market Index
BVSP	Sao Paulo's Stock Market Index
EU	MSCI European Index
EM	MSCI Emerging Markets Index

5. Dataset

The dataset used for the experiments is the "Istanbul Stock Exchange", created and used by Akbilgic et al. (2014) in their work, and made available on the UCI repository (Lichman, 2013). The dataset contains 536 data points, each representing a day and composed of nine floating point numbers indicating daily returns between January 5, 2009, to February 22, 2011, for the market indexes in Table 1.

The data was processed by the authors just so the days on which the Turkish stock exchange was closed were removed. Missing values on the time series indexes were replaced by their immediate valid past value. Akbilgic et al. (2014) reported the accuracy and the cumulative return obtained by trading ISE100 based on the predictions of their model, a Hybrid RBF Neural Network. ISE100_TL and ISE100_USD represent the same market index (ISE100), but one of them computed with respect to US Dollars and the other one with respect to the Turkish Lira. ISE100_TL was used in this work exclusively to generate results that could be compared to the accuracy and cumulative return reported in Akbilgic et al. (2014). As an abstraction, in this work, we assume the existence of a portfolio that tracks the market indexes and can be rebalanced at will, as to effectively function as a single financial instrument, since Market Indexes cannot be traded.

6. Experimental protocol

Our protocol was designed to compare the trend accuracy and the theoretical cumulative return obtained by trading based on the classification from 4 different classifiers, using 1-day lagged values of the raw time series in one case and the 1-day lagged values of the components obtained with EEMD. Training and testing the models consisted of an interactive process that simulates daily trading. The initial training set consisted of the first 250 days, and the test set a single data point, the 251th day. After each iteration, the day used as the test set gets added to the training set, and the next day with respect to the last test day is used as the new test set, up until the 450th day is used as the test set. The initial training size, 250 days, and final testing day, 450, were chosen just so the results of this work could be compared to the results published by the authors of the dataset.

Each data pre-processing step must be done within the training and testing iterations as to not add look-ahead bias; as explained previously, decomposing the time series with EEMD as a pre-processing step would add a flaw to the protocol. After the decomposition, each component is normalized.

From a high level perspective, our protocol is described on Fig. 3. Algorithms 3 and 4 contain a detailed description of the protocol used for this work:

On Algorithm 4, despite the training set and the testing set being concatenated before the decomposition, this does not add look-ahead bias to the protocol. The only information we do not possess at the end of each day is the class associated to the most current

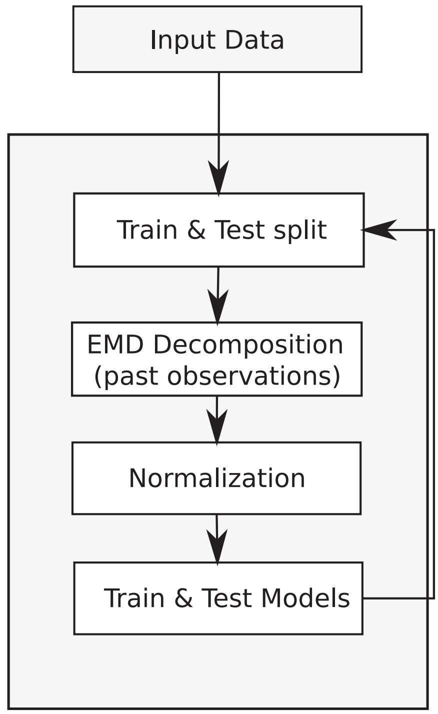

Fig. 3. Protocol without look-ahead bias.

Algorithm 3 High level protocol.

```
Require: data_set, nFeats, nIMFs
Ensure: results
results = []
for each combination of parameters do
   Get all feature combinations with nFeats features
   for Each feature combination do
      models = LinearSVM, RBF-SVM, RF and LogRegress
      lag_dataset = CreateLaggedDataset(dataset)
      results.append(RunTrainTest(0, 250, 450, lag_dataset,
      nIMFs, models))
   end for
end for
return results
```

Algorithm 4 RunTrainTest.

```
Require: start_train, end_train, FINAL_DAY, dataset, nIMFs, models
Ensure: results
results = []
while end_train < FINAL_DAY do
   train_set = dataset[start_train:end_train - 1]
   test_set = dataset[end_train]
   imfs = EEMD(train_set + test_set, nIMFs)
   norm_Imfs = Normalize(imfs)
   imfs_train_set = norm_Imfs[start_train:end_train - 1]
   imfs_test_set = norm_Imfs[end_train]
   for each model in models do
      Train model 4-fold cross validation on imfs_train_set
      Test model on imfs_test_set
   end for
   end_train += 1
end while
for each model in models do
   results.append(model.results)
end for
return results
```

observation, which is defined by whether the price will go up or down by the end of the day tomorrow. We also need to consider that the test set is always composed of a single data point, which is computed with yesterday's and today's closing price; this is all data we possess. This protocol maps to a real situation where the training set is composed of all the historical observations we have but the last data point of the time series, which is the percent change value computed between yesterday's closing price and today's closing price. The predictive models are trained and tested after the markets close, so we can predict, at the end of each day, what will happen tomorrow.

In order to evaluate the effectiveness of EEMD alone, we do an exhaustive search over all the possible combinations of market indexes by training and testing the models with each combination and storing the results for later comparison. The 1-day lagged values of the target index was also part of the feature pool.

The parameters *S*_number and number of siftings of EEMD were set to 4 and 50 respectively, the default value in the library used for the tests (Luukko, Helske, & Räsänen, 2016). These parameters were not changed during the tests because preliminary simulations have shown that the impact they had on the IMFs generated produce were negligible. The values of noise strength and the size of the ensemble were set in accordance to the guidelines presented on EEMD's seminal paper (Wu & Huang, 2009): The noise strength amplitude was set to be 0.2 of the standard deviation of the input signal, and the size of the ensemble, *K*, was set at 250. Despite being parameters that do affect the decomposition accuracy, results in the literature indicate that increasing noise amplitudes and ensemble size do not alter the decomposition considerably as long as the added noise has moderate amplitude and the ensemble is large enough (Wu & Huang, 2009). For the reader's reference, it should be noted that recent studies proposing techniques to improve the selection of these parameters exist and shown to be effective for other signals, such as a vibration signal from machinery (Du, Liu, Huang, & Li, 2016).

A total of 324 possible combinations of classifiers and features were used to build the pool of models using raw values and EEMD components as input. For each pool, four different classifiers were trained (Linear SVM, RBF-SVM, Logistic Regression and Random Forests) with all the possible combinations of 4 or more indexes in the feature vector. For the cases where EEMD was used to extract the components, the number of features is multiplied by eight since this is the number of components extracted from the indexes, which matches the theoretical number of components extracted from time series of size n, $log_2 n$ (Wu & Huang, 2009).

The metrics used to compare the results among the different models were the Trend accuracy and the Cumulative Return, the later being a popular metric to compare the performance of different financial instruments. The Trend accuracy is given by Eq. (1):

$$\text{TrendAcc} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

where the True Positive (TP) and True Negative (TN) are the number of correct predictions for up trends and down trends, respectively. The denominator sums up to the total number of predictions performed by the model.

The standard formula for cumulative return (Rc) between days *a* and *b* is given by Eq. (2)

$$\text{Rc} = \prod_{i=a}^{b} \left(1 + \frac{P_{i+1} - P_i}{P_i}\right) \tag{2}$$

where P_i is the closing price of the financial instrument at the *i*th day. However, to compute the cumulative return taking in consideration the accuracy of the predictions, Eq. (2) needs to be slightly

Table 2
Comparing study results to the one in the literature.

	ISE_TL					
	Raw		EEMD		EEMD*	
	Accuracy	$R_c(\%)$	Accuracy	$R_c(\%)$	Accuracy	$R_c(\%)$
Lin SVM	0.660	246.65	0.600	**256.70**	0.660	331.25
RBF SVM	0.655	**271.53**	**0.620**	246.61	0.675	343.58
Log. Reg.	0.650	233.18	0.600	228.85	**0.685**	**360.44**
RF	0.615	207.17	0.560	162.47	0.615	217.25
HRBF-NN (Akbilgic et al., 2014)	**0.68**	202	–		–	–

different:

$$\mathrm{Rc} = \prod_{i=a}^{b} \begin{cases} 1 + abs(\frac{P_{i+t}-P_i}{P_i}) \mapsto TP & \text{and} \quad TN \\ 1 - abs(\frac{P_{i+t}-P_i}{P_i}) \mapsto FP & \text{and} \quad FN \end{cases} \tag{3}$$

In this scenario, we consider a hypothetical situation where we are able to short or long any one of these market indexes for a day, with the profit from this transaction being the full percentage change from today's to tomorrow's closing price.

The Cumulative Return is a specially important metric. Despite having a high accuracy, a specific model can present a lower cumulative return if it is doesn't perform well in detecting strong up or down movements. These two metrics were also used in the study that introduced the dataset used in this study (Akbilgic et al., 2014), which allows us to use the reported results as another data point.

7. Results

As the benchmarks for Accuracy and Cumulative Return, we use the coin flip probability of predicting accurately the trend and a buy-and-hold strategy, respectively. The buy-and-hold (BH) return for a financial instrument after t days is simply defined by Eq. (4):

$$\mathrm{BH}_i = 1 + \frac{P_{i+t} - P_i}{P_i}, \tag{4}$$

where P_i is the closing price of the financial instrument at the ith day. A return larger than 1 implies earning with respect to the initial capital, and a loss when the value is smaller.

The best results were reported with respect to the model, accuracy, cumulative return and input representation in Tables 3 and 4, and in Figs. 4 and 5. The bar charts show the best results across all models, and on the tables we report the best results per model. The results obtained with EEMD used as a pre-processing step (with look-ahead bias) are referenced by the label EEMD*. We also present a comparison between the results obtained by the Akbilgic et al. using a model of their authorship (HRBF-NN) and our algorithm using EEMD on Table 2. The values in bold represent the best accuracy and cumulative return obtained by each model using the best combination of features found via exhaustive search, as explained previously.

For the sake of brevity, the list of features per model was not reported, but worth noting is the fact that the best number of market indexes for the models using the raw values were, on the large majority of the cases, larger than for the models using EEMD components. A single index added to the list of feature actually adds 8 components to the feature vector due to the decomposition, so this difference might be explained by the curse of dimensionality. For the results reported under EEMD*, there is a strong tendency of the best results being obtained with all the features, which indicates the look-ahead bias embedded useful information about future behavior in every market index decomposed.

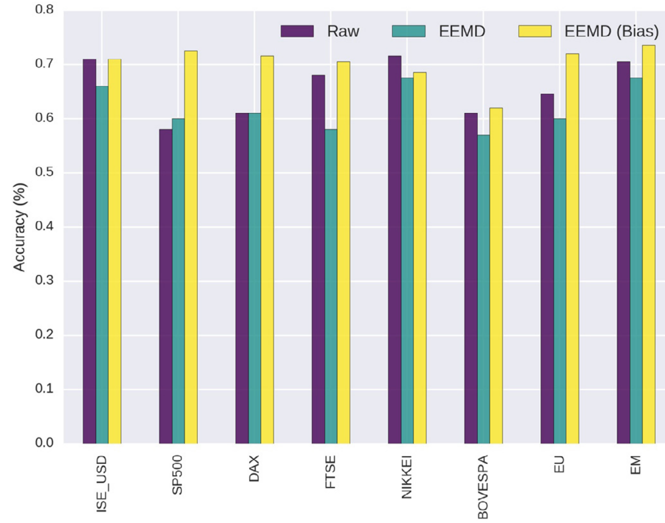

Fig. 4. Best accuracy per market index.

Fig. 5. Best Cumulative Returns (R_c) per market index.

The models, whether using the raw values or the EEMD components, were able to consistently beat the buy and hold strategy for the marked indexes used in this study, as shown on Table 4.

The results however show the difference that exists between the performance of the models when using a proper protocol for the tests and one with look-ahead bias added to it. In the large majority of the tests the models using the components extracted from the entire time series with EEMD as a pre-processing (tests with look-ahead bias) beat all the other models.

In contrast to these results, a protocol created to eliminate the look-ahead bias from the application of EEMD tells a different story. Despite the models using the EEMD components achieving a considerable difference in the cumulative return with respect to the original study, the best performing models when predicting the trend direction and also with respect to the cumulative return were, in their majority, models using the raw percent change values instead of the components generated with EEMD. For the ISE_USD, DAX, FTSE, NIKKEI, BOVESPA, EU and EM indexes, the best models were the models using the raw percent change values instead of the values extracted with EEMD. For the S&P500 market index, the models trained with the EEMD components performed consistently better than the models trained with the raw values.

Table 3
Best accuracy results per predictive model (with best parameters & best feature combination).

	ISE_USD			SP500			DAX			FTSE		
	Raw	EEMD	EEMD*	Raw	EEMD	EEMD*	Raw	EEMD	EEMD*	Raw	EEMD	EEMD*
Lin. SVM	**0.710**	0.660	0.700	0.545	**0.600**	0.725	0.600	**0.610**	0.710	0.665	0.575	0.705
RBF SVM	0.705	0.640	0.710	0.550	0.595	0.715	0.590	0.595	0.715	0.635	0.560	0.635
Log. Reg.	0.700	0.625	0.700	0.535	0.590	0.715	0.600	**0.610**	0.715	0.665	0.570	0.685
RF	0.635	0.620	0.640	0.580	0.580	0.640	**0.610**	0.600	0.605	**0.68**	0.580	0.565
	NIKKEI			BOVESPA			EU			EM		
	Raw	EEMD	EEMD*	Raw	EEMD	EEMD*	Raw	EEMD	EEMD*	Raw	EEMD	EEMD*
Lin. SVM	**0.715**	0.675	0.675	0.500	0.510	0.600	0.630	0.590	0.705	**0.705**	0.675	0.715
RBF SVM	0.705	0.660	0.675	0.505	0.520	0.620	0.595	0.590	0.700	0.695	0.650	0.735
Log Reg	0.710	0.660	0.685	0.505	0.515	0.600	0.630	0.590	0.720	**0.705**	0.655	0.715
RF	0.680	0.635	0.600	**0.610**	0.570	0.580	**0.645**	0.600	0.615	0.695	0.650	0.645

Table 4
Best Cumulative Return results per predictive model (with best parameters & best feature combination).

	ISE_USD			SP500			DAX			FTSE		
	Raw	EEMD	EEMD*	Raw	EEMD	EEMD*	Raw	EEMD	EEMD*	Raw	EEMD	EEMD*
Lin. SVM	481.9	317.6	541.0	108.9	120.1	319.6	**203.3**	154.2	332.2	**224.7**	147.0	290.8
RBF SVM	492.1	314.0	570.0	116.2	124.7	306.6	190.3	136.7	295.8	196.4	145.8	262.9
Log. Reg.	**497.6**	311.4	553.7	104.6	133.5	297.4	202.7	168.7	316.4	214.8	146.4	277.1
RF	276.5	279.9	309.8	128.2	**146.0**	230.5	177.5	157.2	204.3	179.4	154.9	196.2
BH	132.7	132.7	132.7	103.0	103.0	103.0	106.1	106.1	106.1	103.0	103.0	103.0
	NIKKEI			BOVESPA			EU			EM		
	Raw	EEMD	EEMD*	Raw	EEMD	EEMD*	Raw	EEMD	EEMD*	Raw	EEMD	EEMD*
Lin. SVM	**408.3**	294.9	376.9	106.7	100.1	240.6	**210.3**	150.6	322.8	**193.5**	180.3	207.2
RBF SVM	391.7	311.3	369.2	116.0	104.1	270.0	201.5	141.0	290.1	185.0	176.5	216.7
Log. Reg.	390.0	299.7	373.1	107.1	100.7	246.7	202.4	144.8	335.8	189.9	169.6	208.7
RF	297.7	217.0	223.3	**155.6**	142.1	180.7	174.2	144.1	206.9	189.8	157.4	180.6
BH	87.5	87.5	87.5	100.7	100.7	100.7	99.2	99.2	99.2	106.5	106.52	106.5

In terms of the results obtained by each individual model with the raw values and the EEMD components without look-ahead bias, it is interesting to note that the Linear SVM was the best performing model when predicting the trend of 5 out of 8 market indexes contained in the dataset. Random Forests, despite achieving good accuracy results, are the worst performing model in general regarding cumulative returns, with less accurate predictions when detecting large movements.

8. Conclusion

The objective of this study was to evaluate the effectiveness of Ensemble Empirical Mode Decomposition at generating features to be used as trend predictors in the Istanbul Market Index dataset (Akbilgic et al., 2014). Our testing protocol, in contrast to protocols defined in other studies found in the literature, does not add look-ahead bias through the use of EEMD.

The results obtained with the use of our protocol indicate, differently to the results presented in the literature, that the models trained with the EEMD components do not outperform, in terms of Accuracy and Cumulative Return, the models trained with the raw percent values for the majority of the market indexes contained in the dataset used for the study. The exception to this was S&P500, which was an isolated case.

On the other hand, the models trained with the EEMD components extracted as a pre-processing step from the entire time series outperformed all the other models, indicating that the look-ahead bias heavily affects the accuracy, and ultimately the cumulative return, of the models by generating components that encode information about the future on past data points. These results reinforce the need to be extremely careful when using techniques that might make use of values that are not contained in the training set.

These results, however, do not necessarily mean that these components are not useful as predictors. They might not be better predictors on their own when compared to the raw percentage values, but these components might be useful if used in conjunction with the raw values. Additionally, the models did perform better when predicting the trend of S&P500, indicating they may be useful for prediction tasks on specific time series.

Finally, as future work, the authors intend to look further into the impact caused by the rough decomposition at the end of the time series when using EMD/EEMD. The impossibility of extrapolating the very last value of the series to subsequent points generate a crude decomposition at the end, which might actually be one of the main culprits for the drastic reduction in accuracy obtained by the classification models.

References

Akbilgic, O., Bozdogan, H., & Balaban, M. E. (2014). A novel hybrid RBF neural networks model as a forecaster. *Statistics and Computing, 24*(3), 365–375.

Al-Hnaity, B., & Abbod, M. (2015). A novel hybrid ensemble model to predict FTSE100 index by combining neural network and EEMD. In *Control conference (ECC), 2015 European* (pp. 3021–3028). IEEE.

Du, S.-C., Liu, T., Huang, D.-L., & Li, G.-L. (2016). An optimal Ensemble Empirical Mode Decomposition method for vibration signal decomposition. *Journal of Vibration and Acoustics*.

Fenghua, W., Jihong, X., Zhifang, H., & Xu, G. (2014). Stock price prediction based on SSA and SVM. *Procedia Computer Science, 31*, 625–631.

Ghodsi, M., Hassani, H., & Sanei, S. (2010). Extracting fetal heart signal from noisy maternal ECG by singular spectrum analysis. *Journal of Statistics and its Interface, Special Issue on the Application of SSA, 3*(3), 399–411.

Hsu, M.-W., Lessmann, S., Sung, M.-C., Ma, T., & Johnson, J. E. (2016). Bridging the divide in financial market forecasting: machine learners vs. financial economists. *Expert Systems with Applications, 61*, 215–234.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., … Liu, H. H. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. In *Proceedings of the royal society of london a: Mathematical, physical and engineering sciences: 454* (pp. 903–995). The Royal Society.

Huang, N. E., Wu, M.-L. C., Long, S. R., Shen, S. S., Qu, W., Gloersen, P., & Fan, K. L. (2003). A confidence limit for the empirical mode decomposition and Hilbert spectral analysis. In *Proceedings of the royal society of london a: Mathematical, physical and engineering sciences: 459* (pp. 2317–2345). The Royal Society.

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*(1), 307–319.

Kumar, M., & Thenmozhi, M. (2006). Forecasting stock index movement: A comparison of support vector machines and random forest. *Indian institute of capital markets 9th capital markets conference paper*.

Lei, Y., Lin, J., He, Z., & Zuo, M. J. (2013). A review on empirical mode decomposition in fault diagnosis of rotating machinery. *Mechanical Systems and Signal Processing, 35*(1), 108–126.

Lichman, M. (2013). UCI machine learning repository. http://archive.ics.uci.edu/ml.

Luukko, P., Helske, J., & Räsänen, E. (2016). Introducing libeemd: A program package for performing the Ensemble Empirical Mode Decomposition. *Computational Statistics, 31*(2), 545–557.

Mahfoud, S., & Mani, G. (1996). Financial forecasting using genetic algorithms. *Applied Artificial Intelligence, 10*(6), 543–566.

Mikosch, T., & Stărică, C. (2004). Nonstationarities in financial time series, the long-range dependence, and the IGARCH effects. *Review of Economics and Statistics, 86*(1), 378–390.

Sapankevych, N. I., & Sankar, R. (2009). Time series prediction using support vector machines: a survey. *IEEE Computational Intelligence Magazine, 4*(2), 24–38.

Tang, L., Wang, S., & Yu, L. (2011). EEMD-LSSVR-based decomposition-and-ensemble methodology with application to nuclear energy consumption forecasting. In *Computational sciences and optimization (CSO), 2011 fourth international joint conference on* (pp. 589–593). IEEE.

Vautard, R., Yiou, P., & Ghil, M. (1992). Singular-spectrum analysis: A toolkit for short, noisy chaotic signals. *Physica D: Nonlinear Phenomena, 58*(1), 95–126.

Wang, T., Zhang, M., Yu, Q., & Zhang, H. (2012). Comparing the applications of EMD and EEMD on time–frequency analysis of seismic signal. *Journal of Applied Geophysics, 83*, 29–34.

Wu, Z., & Huang, N. E. (2009). Ensemble Empirical Mode Decomposition: A noise-assisted data analysis method. *Advances in adaptive data analysis, 1*(01), 1–41.

Xiong, T., Bao, Y., Hu, Z., Zhang, R., & Zhang, J. (2011). Hybrid decomposition and ensemble framework for stock price forecasting: A comparative study. *Advances in Adaptive Data Analysis, 3*(04), 447–482.

Yu, L., Wang, S., & Lai, K. K. (2008). Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. *Energy Economics, 30*(5), 2623–2635.