Applied Soft Computing Journal 93 (2020) 106422

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

Predicting next day direction of stock price movement using machine learning methods with persistent homology: Evidence from Kuala Lumpur Stock Exchange

Mohd Sabri Ismail *, Mohd Salmi Md Noorani, Munira Ismail, Fatimah Abdul Razak, Mohd Almie Alias

Department of Mathematical Sciences, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

ARTICLE INFO

Article history:
Received 4 July 2019
Received in revised form 25 March 2020
Accepted 20 May 2020
Available online 26 May 2020

Keywords:
Stock price movement prediction
Machine learning methods
Persistent homology
Technical indicators

ABSTRACT

Predicting direction of stock price movement is notably important to provide a better guidance to assist market participants in making their investment decisions. This study presents a hybrid method combining machine learning methods with persistent homology to improve the prediction performance. Three stock prices namely Kuala Lumpur Composite Index, Kuala Lumpur Stock Exchange Industrial and Kuala Lumpur Stock Exchange Technology sampled from Kuala Lumpur Stock Exchange are selected for experimental evaluation. In particular, persistent homology was applied to obtain a new and useful input vectors of invariant topological features from returns of these stock prices for further classification task using machine learning methods such as logistic regression, artificial neural network, support vector machine and random forest to predict the next day movement direction of Kuala Lumpur Composite Index. For comparative analysis, we compare the proposed method with others, where the machine learning methods are applied independently on stock returns and also on technical indicators respectively. By using the average of prediction performances and pairwise model comparison method, these two evaluation measures revealed that machine learning methods with persistent homology produced better prediction performance. Our results also demonstrated that the combination of support vector machine with persistent homology generates the best outcome. In general, a combination of machine learning methods with persistent homology is an emerging and promising alternative tool for predicting direction of stock price movement.

1. Introduction

Stock market plays a vital role in a country's economic system and is one of the primary indicators for economic condition [1,2]. Stock market provides opportunities for market participants such as investors, traders or the general public to raise their wealth through stock investments. Nevertheless, precautionary steps need to be taken to tackle market risks where the value of their investment may increase or decrease due to stock market conditions. In particular, determining the future direction of stock price movement is notably important to provide a better guidance in market investment decisions for high yield financial returns and to hedge against market risks [3]. Therefore, the search for models with accurate prediction of stock price has been a highly researched topic. Predicting direction of stock price movement is regarded as one of the challenging tasks in the financial time series analysis and machine learning [4]. The challenge is mainly due to stock market's behaviors which are inherently complex, noisy, evolutionary, non-linear and deterministically chaotic [5,6]. All these behaviors are influenced by many interacting factors such as movement of other stock prices, political events, economic conditions, government policies, trader's expectations and catastrophe or war [7,8].

Previous studies have demonstrated that methods of machine learning have huge potential to tackle the modeling and prediction in nonlinear and multivariate financial data [9]. Essentially, two classes of machine learning methods have been introduced: (1) single base models such as logistic regression (LR) [10], artificial neural network (ANN) [11,12], support vector machine (SVM) [13–15], *k*-nearest neighbor (KNN) [16], Naive Bayes model (NB) [17], etc. and (2) ensemble or multiple base models such as random forest (RF) [18,19], extreme gradient boosting (XGB) [20,21] and so on. In the domain of predicting direction of stock price movement, the machine learning methods that have been applied in numerous studies and used for the prediction are LR, ANN,

* Corresponding author.
E-mail addresses: p94450@siswa.ukm.edu.my (M.S. Ismail), msn@ukm.edu.my (M.S. Md Noorani), munira@ukm.edu.my (M. Ismail), fatima84@ukm.edu.my (F. Abdul Razak), mohdalmie@ukm.edu.my (M.A. Alias).

https://doi.org/10.1016/j.asoc.2020.106422
1568-4946/

SVM and RF [9,10,22]. These machine learning methods which integrate artificial intelligence systems, learn from training data by extracting nonlinear patterns between its input and output. Afterwards, these methods use the extracted patterns to predict the output for testing or new data.

In predicting the direction of stock price movement using machine learning methods, finding suitable features representation is imperative to improve prediction performance. However, consensus has not been reached on the most effective information or features to be used as the input for those machine learning methods. In practice, particularly for the next day prediction, previous stock prices or some other variant of these prices were used by those machine learning methods to perform short-term prediction [23–27]. This has been driven by the idea that if the market can be predicted, then the previous stock prices or features extracted from them (e.g. technical indicators based on technical analysis [13,28]) should explain some variation in prices or returns [29]. Besides, macroeconomic factors rooting from fundamental analysis are also important features for the prediction [30,31]. Unfortunately, fundamental analysis is inadequate for short-term predictions [32]. In addition, some features extracted from various sources of financial and economic data based on text analysis or sentiment analysis are also applied in prediction of stock price movement [33–35].

Recently, an emerging and fast-growing field called persistent homology (PH) has proposed the idea that invariant topological features of high dimensional and complex data are capable to provide relevant and useful patterns on the behaviors of the underlying topological space represented by the data (see [36] and references therein for a survey on PH). It has been demonstrated that the multiscale descriptor of topological features obtained from PH such as persistent Betti numbers are very useful in finding relevant patterns from data of possible complex and chaotic dynamical systems. Among successful discoveries of PH include analysis patterns of complex image data of turbulent flows [37], analysis of microstructure data [38], discovery of a subgroup of breast cancers [39], detection and quantification of periodic patterns in chaotic data [40,41] and understanding topological patterns of chaotic attractors in phase space [42]. With respect to financial data, PH had also been developed to detect a growing systemic risk in United States stock market [43,44] and in cryptocurrencies [45].

Remarkably, some machine learning methods have been employed to learn and predict with PH. For example, SVM with PH has shown favorable outcomes such as inferencing imaging datasets of Alzheimer's disease [46], detecting conformational changes between closed and open forms of maltose-binding protein [47] and recognizing atmospheric river patterns in large climate datasets [48], while combination of LR with PH have also been applied in image analysis [49]. Additionally, several machine learning methods namely Gaussian-based decision tree, decision tree, RF and SVM with PH were found to be able to improve accuracy in predicting time series data [50]. To the best of our knowledge, such combination has not been explored for financial time series data especially to predict the next day direction of stock price movement.

The objective of this paper is to explore combination of the commonly used machine learning methods – LR, ANN, SVM and RF – with PH as an emerging potential tool to improve prediction performance. In particular, we aim to demonstrate that the integration of PH in these machine learning methods will provide better prediction results compared to using the machine learning methods independently on stock returns and also on technical indicators. Therefore, PH is applied to provide a good grasp on the evolution pattern of the next day direction of stock price movement and to enhance prediction using those patterns. Specifically, we investigate persistent Betti numbers obtained from stock returns using PH to understand the evolution pattern of the next day direction of Kuala Lumpur Composite Index movement and to create new alternative input vectors of topological features for further classification task using machine learning methods mentioned above. The remaining portion of this paper is organized as follows: Section 2 briefs about the research data, Section 3 provides details on the proposed method, Section 4 presents the results, Section 5 discusses the results and Section 6 wraps the conclusion and provides suggestions for future works.

2. Data

In this study, we collect three daily closing stock prices for Kuala Lumpur Composite Index (KLCI), Kuala Lumpur Stock Exchange Industrial (KLSE IND) and Kuala Lumpur Stock Exchange Technology (KLSE TEC) sampled from Kuala Lumpur Stock Exchange (KLSE). KLSE (also known as Bursa Malaysia) is an emerging market in Malaysia and one of the largest markets in South East Asia in terms of its domestic market capitalization [51]. In Malaysia, KLSE assist over 900 companies to raise capital across 60 economic activities. On another note, KLCI, KLSE IND and KLSE TEC are the main indicator indexes of KLSE which measure the performance of leading companies in all sectors, industrial sector and technology sector respectively. All the historical data for these daily closing stock prices are obtained from DataStream, dating between 15/05/2000 to 27/03/2018 totaling to 4662 trading days. In this prediction exercise, we transform all daily closing stock price to stock return using daily log-returns, formulated as $x(t) = \ln(p(t)/p(t-1))$, where $p(t)$ and $x(t)$ are the daily closing stock price and the corresponding stock return for trading day t respectively. This transformation helps to generate an effective detrending of the original daily closing stock prices [52]. Fig. 1 shows the daily closing price of KLCI and the stock returns of KLCI, KLSE IND and KLSE TEC at those available trading days respectively. As shown in this figure, it is a challenge to obtain an explicit financial time series model which describes the underlying non-linear relationship between these stock returns in predicting the next day direction of KLCI movement. However, machine learning method offers promising alternative tool that is capable to predict direction of KLCI movement [53–55]. In addition, [44] demonstrated that persistent homology can be employed on main stock returns to detect a growing systemic risk in US stock market as highlighted in the earlier introduction. This indicates that PH has huge potential in extracting useful patterns of stock returns in performing classification task using machine learning method. Therefore, the initiative to explore the combination of machine learning method with PH is deemed sensible.

3. Method

3.1. Stock returns

At the inception of this study, we have a collection of three one-dimensional financial time series, which are stock returns of Kuala Lumpur Composite Index (KLCI), Kuala Lumpur Stock Exchange Industrial (KLSE IND) and Kuala Lumpur Stock Exchange Technology (KLSE TEC) as illustrated in Figs. 1(b), (c) and (d) respectively. The collection can be denoted as $\{SR_i | i = 1, 2, 3\}$, where SR_1, SR_2 and SR_3 are stock returns of KLCI, KLSE IND and KLSE TEC accordingly. For each $i = 1, 2, 3$, SR_i is a set of 4661 points, that is $SR_i = \{x_i(t) \in \mathbb{R} | t = 1, 2, \ldots, 4661\}$, where $x_i(t)$ is a value of SR_i at the trading day t. Collectively, we combine all these SR_i in increasing manner of i to obtain a new three-dimensional time series data, denoted as $X =$

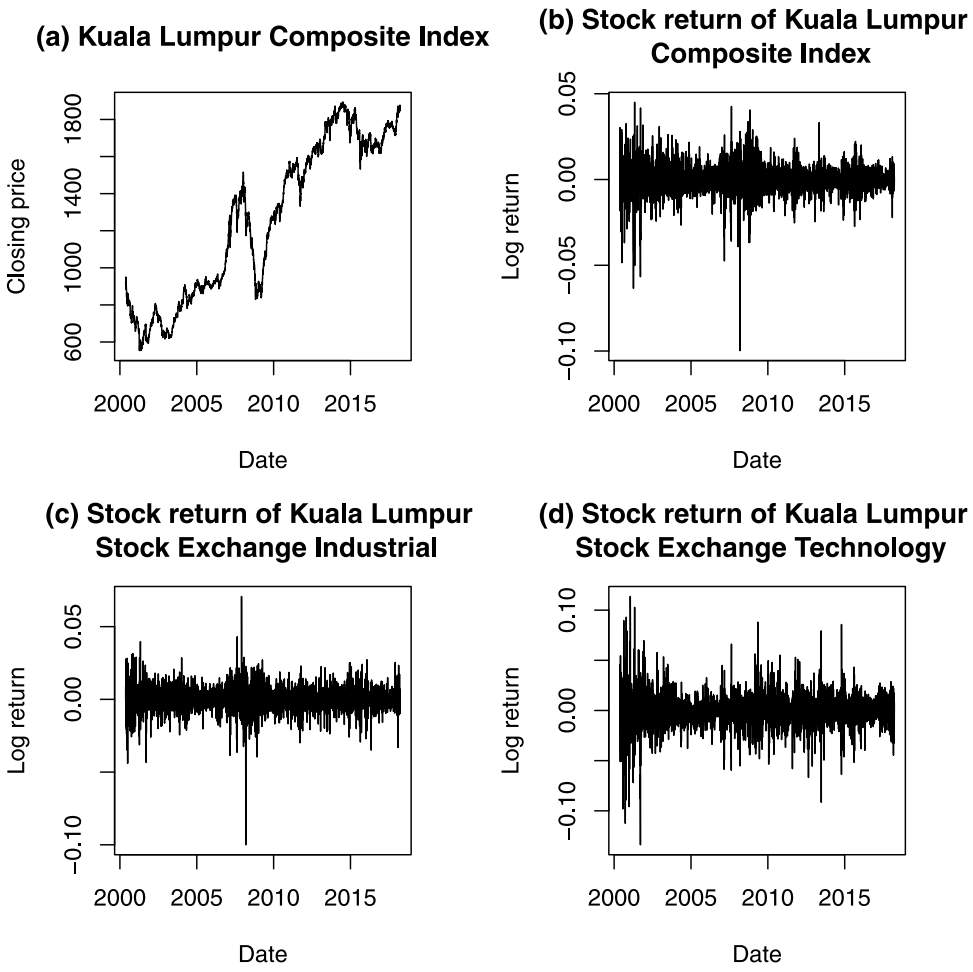

Fig. 1. From (a) to (d) are the daily closing price of Kuala Lumpur Composite Index (KLCI) and the stock returns of KLCI, Kuala Lumpur Stock Exchange Industrial (KLSE IND) and Kuala Lumpur Stock Exchange Technology (KLSE TEC) respectively.

$\left\{y(t) \in \mathbb{R}^3 | t = 1, 2, \ldots, 4661\right\}$ where vector of $y(t) = (x_1(t), x_2(t), x_3(t))$ at the trading day t. If set X is written in matrix form, the set X becomes a 4661×3 matrix as illustrated below:

$$X = \begin{bmatrix} x_1(1) & x_2(1) & x_3(1) \\ x_1(2) & x_2(2) & x_3(2) \\ \vdots & \vdots & \vdots \\ x_1(4661) & x_2(4661) & x_3(4661) \end{bmatrix}.$$

Each column of the matrix X is a feature given by values of SR_i for every $i = 1, 2, 3$. Typically, one could use X as input vectors for machine learning method to learn and predict the next day direction of KLCI movement. In this study, we perform classification task using machine learning methods – logistic regression (LR), artificial neural network (ANN), support vector machine (SVM) and random forest (RF) – independently on X as a comparison case to the proposed combination of those machine learning methods with persistent homology (PH).

3.2. Technical indicators

As another comparison case to PH, we compute seven technical indicators from the daily closing price of KLCI. The seven selected technical indicators and their formulae are summarized in Table 1. These selected technical indicators are also used as input vectors for those machine learning methods (LR, ANN, SVM and RF) to learn and predict the next day direction of KLCI movement.

3.3. Daily sliding window

Prior to application of PH, daily sliding window method [42,44] is performed to segment the X (stock returns as described in Section 3.1) to obtain a collection of time windows of size w. Herein, the parameter value w is called window size which refers to the chosen length for each time window along the whole time series X. For example, by continuously choosing $w = 60$, the time series X is then segmented into a collection of 4612 time windows of size 60, denoted as $\left\{TW^{60}(j) \subset X | j = 60, 61, \ldots, 4661\right\}$, where time window of size 60 at the trading day j is $TW^{60}(j) = \left\{y(t) \in \mathbb{R}^3 | t = j-61, j-62, \ldots, j\right\}$. Note that trading day j starts from 60 to ensure only the corresponding past vectors are used for future estimation. For all degree j, $TW^{60}(j)$ is a point cloud data (PCD) which consists of 60 vectors. In matrix form, the respective PCD can be represented as a 60×3 matrix as shown below:

$$TW^{60}(j) = \begin{bmatrix} x_1(j-61) & x_2(j-61) & x_3(j-61) \\ x_1(j-62) & x_2(j-62) & x_3(j-62) \\ \vdots & \vdots & \vdots \\ x_1(j) & x_2(j) & x_3(j) \end{bmatrix}$$

for each $j = 60, 61, \ldots, 4661$.

3.4. Persistent homology

Subsequently, PH is applied to compute persistent Betti numbers of topological features that persist across a filtration of

Table 1
Selected technical indicators and their formulae.

Name of indicator	Formula
Simple 10-day moving average (MA_{10})	$MA_{10}(t) = \frac{p(t-9)+p(t-8)+\cdots+p(t)}{10}$
Weighted 10-day moving average (WMA_{10})	$WMA_{10}(t) = \frac{(10)p(t-9)+(9)p(t-8)+\cdots+(1)p(t)}{10+9+\cdots+1}$
Momentum (M)	$M(t) = p(t) - p(t-9)$
Stochastic $K\%$ (SK)	$SK(t) = \left(\frac{p(t)-lp(t)}{hp(t)-lp(t)} \times 100\right)\%$
Stochastic $D\%$ (SD)	$SD(t) = \frac{SK(t-9)+SK(t-8)+\cdots+SK(t)}{10}\%$
Relative strength index (RSI)	$RSI(t) = 100 - \frac{100}{1+\frac{\left(\left(\sum_{i=0}^{9} UP(t-i)\right)/10\right)}{\left(\left(\sum_{i=0}^{9} DW(t-i)\right)/10\right)}}$
Moving average converge diverge ($MACD_{10}$)	$MACD_{10}(t) = (\alpha)\,DIFF(t) + (1-\alpha)MACD_{10}(t-1)$

$p(t)$ is the daily closing KLCI stock price at trading day t, $lp(t) = min\{p(t-9), p(t-8), \ldots, p(t)\}$, $hp(t) = max\{p(t-9), p(t-8), \ldots, p(t)\}$, $UP(t)$ means upward daily price change at trading day t, $DW(t)$ means downward daily price change at trading day t, $\alpha = 2/11$, $DIFF(t) = EMA_{12}(t) - EMA_{26}(t)$, $EMA_k(t) = (\beta_k)p(t) + (1-\beta_k)EMA_k(t-1)$, $\beta_k = 2/(k+1)$, the first $EMA_k(t)$ which at trading day $t = k$ is $EMA_k(k) = \left(\sum_{t=1}^{k} p(t)\right)/k$ and the first $MACD_{10}(t)$ which at trading day $t = 35$ is $MACD_{10}(35) = \left(\sum_{t=26}^{35} DIFF(t)\right)/10$ respectively.

simplicial complexes built on top of $TW^{60}(j)$. In short, a simplicial complex is one abstract triangulated structure built on top of $TW^{60}(j)$ to approximate the underlying topological space which lies upon the observed vectors of accumulated $TW^{60}(j)$. A single simplicial complex is built by combining together some k-dimensional simplexes at their commonly shared k-dimensional simplexes. As its basic triangulated building component, 0-dimensional simplex is a vertex, 1-dimensional simplex is an edge, 2-dimensional simplex is a triangular face, 3-dimensional simplex is a solid tetrahedron and so on. The dimension of a simplicial complex is equal to the highest dimension of k-dimensional simplexes that it contains. Fig. 2 provides illustrations for 0 until 3-dimensional simplexes and a single 3-dimensional simplicial complex respectively.

In this study, simplicial complex is constructed using formal definition of Vietoris–Rips complex (also simply called Rips complex). Specifically, Rips complex is a way to construct a simplicial complex on top of $TW^{60}(j)$ using the following rules: (1) all vectors of $TW^{60}(j)$ is considered as a vertex and (2) for a parameter $\varepsilon \geq 0$, k-dimensional simplex (for any dimension of $k \geq 1$) is denoted by $\sigma = \{y(1), y(2), \ldots, y(k+1)\}$ belongs to the Rips complex if and only if for every edge $(y(i), y(j))$, where $1 \leq i < j \leq k+1$, we have $|y(i) - y(j)| \leq \varepsilon$ [36,56]. Equivalently, the parameter ε is also thought as radius of balls centered at each vector of $TW^{60}(j)$ and we can construct the Rips complex using the following ball rules: an edge is formed by connecting two ball centers when these two balls intersect each other, a triangular face is formed by connecting three ball centers when these three balls intersect each other and so on for the higher simplexes.

The filtration of Rips complexes is produced by changing the single parameter ε across a set of multiple ascending order, such that $\varepsilon = \varepsilon_0, \varepsilon_1, \ldots, \varepsilon_{max}$, whenever $0 \leq \varepsilon_0 < \varepsilon_1 < \cdots < \varepsilon_{max}$. Therefore, we call the multiple values of ε as evolution parameters which generate a filtration of Rips complexes. Consequently, the filtration of Rips complexes across evolution parameters ε is a nested sequence of $\mathrm{V}\left(TW^{60}(j), \varepsilon_0\right) \subset \mathrm{V}\left(TW^{60}(j), \varepsilon_1\right) \subset \cdots \subset \mathrm{V}\left(TW^{60}(j), \varepsilon_{max}\right)$, whenever $0 \leq \varepsilon_0 < \varepsilon_1 < \cdots < \varepsilon_{max}$. Fig. 3 shows a simple example of a PCD that contains five vectors of $\mathbb{R}^2$ and how a filtration of Rips complexes of this PCD is formed and changed across six evolution parameters using the idea of ball intersection centered at PCD points with radius of the balls as the evolution parameter.

As illustrated in Fig. 3, the invariant topological features that exist in the filtration of Rips complexes are commonly connected components (0-dimensional topological features), holes (1-dimensional topological features), voids (2-dimensional topological features) and so on. Using algebraic topology as its foundation, PH uses kth homology and kth Betti number persistently to track the changing of the k-dimensional topological features that persist in the filtration of Rips complexes. At certain evolution parameter ε, kth homology for the corresponding k-dimensional topological features of the $V\left(TW^{60}(j), \varepsilon\right)$ is a factor group $\mathrm{H}_k\left(V\left(TW^{60}(j), \varepsilon\right)\right) = \mathrm{Z}_k\left(V\left(TW^{60}(j), \varepsilon\right)\right)/\mathrm{B}_k\left(V\left(TW^{60}(j), \varepsilon\right)\right)$, where $\mathrm{Z}_k\left(V\left(TW^{60}(j), \varepsilon\right)\right)$ and $\mathrm{B}_k\left(V\left(TW^{60}(j), \varepsilon\right)\right) \subseteq \mathrm{Z}_k\left(V\left(TW^{60}(j), \varepsilon\right)\right)$ are subgroups of an abelian chain group $\mathrm{C}_k\left(V\left(TW^{60}(j), \varepsilon\right)\right)$ (an algebraic structure used to represent the Rips complex, where its elements are chain, i.e. linear combination of k-dimensional simplexes of Rips complex with coefficients of field $F = \mathbb{Z}_2$) which contain all the chain that represent k-dimensional topological features and k-dimensional topological boundaries respectively.

In brief, kth homology at certain parameter ε, $\mathrm{H}_k\left(V\left(TW^{60}(j), \varepsilon\right)\right)$ is a collection of all k-dimensional topological features which omitted their k-dimensional topological boundaries. Subsequently, kth Betti number is used to compute total number of chain obtained in kth homology. Therefore, kth Betti number at evolution parameter ε is a rank function of $\mathrm{H}_k\left(V\left(TW^{60}(j), \varepsilon\right)\right)$, that is $\beta_k(\varepsilon) = \mathrm{rank}\left(\mathrm{H}_k\left(V\left(TW^{60}(j), \varepsilon\right)\right)\right)$, which is equivalent to total number of all k-dimensional topological features that appear in Rips complex along the parameter ε. Through the filtration of Rips complexes, kth homology and kth Betti number are used to analyze the Rips filtration. Using kth homology persistently on the filtration of Rips complexes, we obtain a nested sequence of kth homologies such as $\mathrm{H}_k\left(\mathrm{V}\left(TW^{60}(j), \varepsilon_0\right)\right) \subset \mathrm{H}_k\left(\mathrm{V}\left(TW^{60}(j), \varepsilon_1\right)\right) \subset \cdots \subset \mathrm{H}_k\left(\mathrm{V}\left(TW^{60}(j), \varepsilon_{max}\right)\right)$, whenever $0 \leq \varepsilon_0 < \varepsilon_1 < \cdots < \varepsilon_{max}$. This implies that a nested sequences of kth persistent Betti numbers computed from the filtration of Rips complexes can also be represented as $\beta_k(\varepsilon_0), \beta_k(\varepsilon_1), \ldots, \beta_k(\varepsilon_{\max})$, whenever $0 \leq \varepsilon_0 < \varepsilon_1 < \cdots < \varepsilon_{max}$. Therefore, kth persistent Betti numbers described how k-dimensional topological features change over the filtration. Remarkably, persistent Betti numbers are perceived as a stable function as small changes of filtration functions imply only small changes of persistent Betti numbers [57]. This has provided PH with stable and reliable persistent Betti numbers.

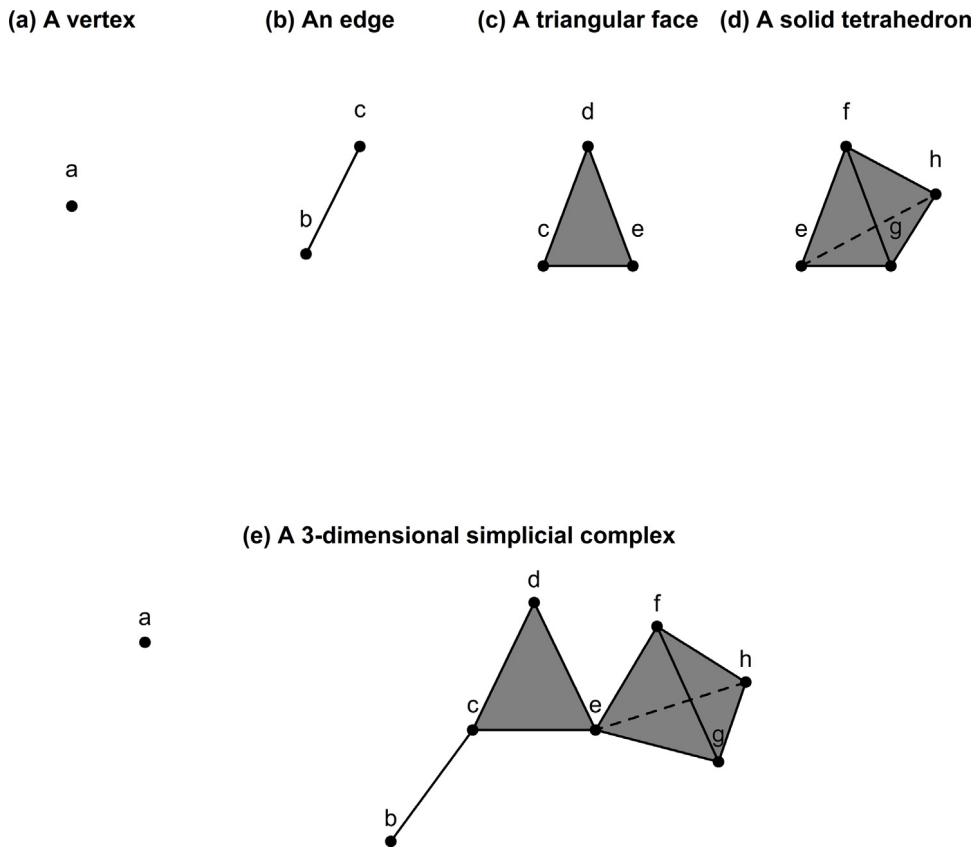

Fig. 2. From (a) to (e) are 0 until 3-dimensional simplexes and a single 3-dimensional simplicial complex respectively. Here, the edge $\{b, c\}$, triangular face $\{c, d, e\}$ and tetrahedron $\{e, f, g, h\}$ are combined at their shared vertices, which are $\{c\}$ and $\{e\}$ respectively to form that simplicial complex.

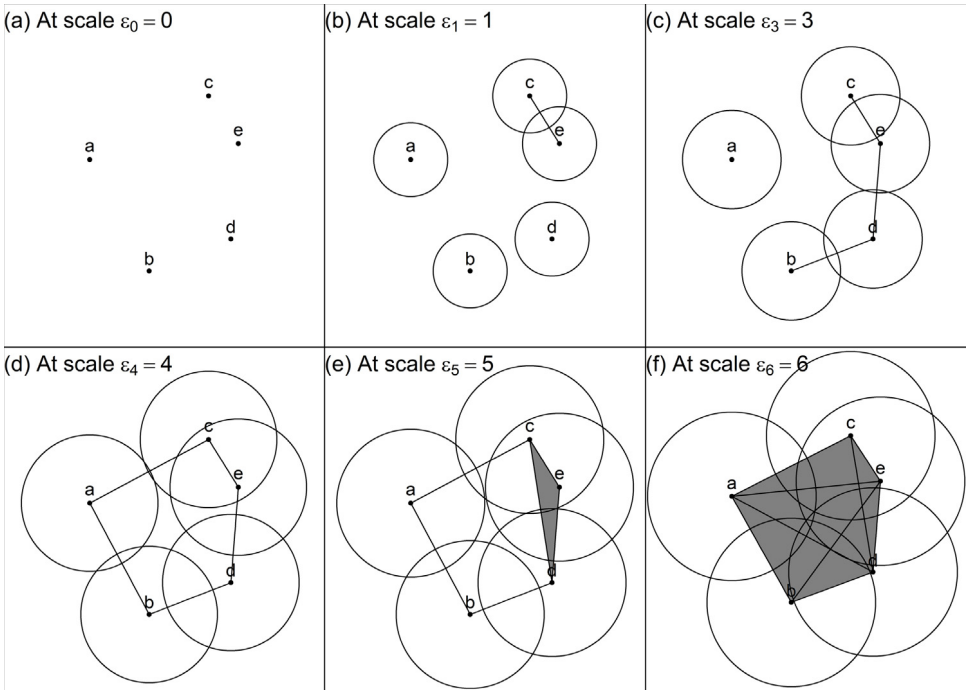

Fig. 3. By using the idea of the intersection of balls centered at PCD points with ball's radius acting as the evolution parameters, (a) to (f) show how the filtration of Rips complexes of this PCD (five vectors of $\mathbb{R}^2$) are formed and changed across six evolution parameters ($\varepsilon = \varepsilon_0, \varepsilon_1, \ldots, \varepsilon_6$). Notice that the simplexes are constantly added and never removed from the previous formations. In the filtration, five connected components appear in (a), four connected components appear in (b), two connected components appear in (c), one connected component with a hole appears in (d) and (e), and one connected component appears in (f) respectively.

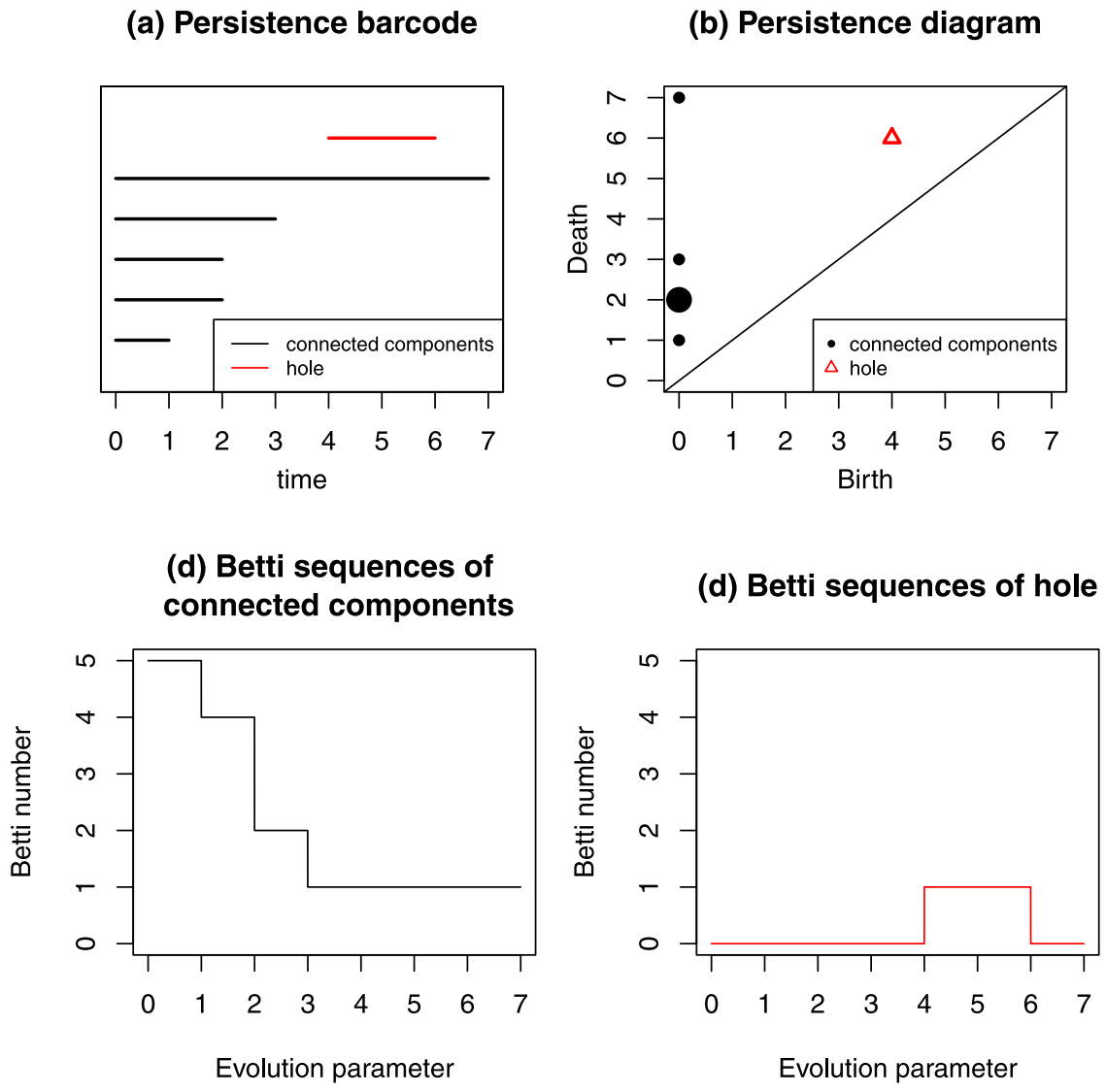

Fig. 4. With regard to the PCD and the corresponding filtration of Rips complexes in Fig. 3, by using PH, we obtain the following illustrations: (a) persistent barcode, (b) persistence diagram (note that the size for point $(0, 2)$ is enlarged two times bigger than the others to indicate there are two connected components which shared similar lifetime), (c) Betti sequences of connected components and (d) Betti sequence of hole respectively.

For the sequence of the kth homologies, there are canonical homeomorphisms that map previous homology to next homology accordingly such that $\mathrm{H}_k\left(\mathrm{V}\left(TW^{60}(j), \varepsilon_0\right)\right) \rightarrow \mathrm{H}_k\left(\mathrm{V}\left(TW^{60}(j), \varepsilon_1\right)\right) \rightarrow \cdots \rightarrow \mathrm{H}_k\left(\mathrm{V}\left(TW^{60}(j), \varepsilon_{max}\right)\right)$, whenever $0 \leq \varepsilon_0 < \varepsilon_1 < \cdots < \varepsilon_{max}$. Consistently, kth persistent Betti numbers are also used to represent the lifetime of a collection of individual k-dimensional topological features. Therefore, a persistent pair of two evolution parameters denoted as $(\varepsilon_b, \varepsilon_d)$ such that $\beta_k(\varepsilon_c) = 1$ whenever $\varepsilon_b \leq \varepsilon_c < \varepsilon_d$ and $\beta_k(\varepsilon_e) = 0$ whenever $\varepsilon_e < \varepsilon_b$ and $\varepsilon_e \geq \varepsilon_d$ is introduced to represent the lifetime. Here, we say that one k-dimensional topological feature is born at ε_b and die at ε_d in the filtration of Rips complexes. Let $\{(\varepsilon_b, \varepsilon_d)_i \,|i = 1, 2, \ldots, n\}$ be a collection of n persistent pairs of the observed k-dimensional topological features that is obtained using PH on $TW^{60}(j)$. There are two most common illustrations popularly used in literature to interpret the collection. One of these is persistent barcode [58–60], which illustrates the collection as n half-closed intervals $[\varepsilon_b, \varepsilon_d)_i$ on the real line; see an example in Fig. 4(a). Another is persistent diagram [61], which illustrates the collection as n birth–death points $(\varepsilon_b, \varepsilon_d)_i \in \mathbb{R}^2$ that lies above the diagonal line in the first quadrant. The diagonal line is treated as part of the persistent diagram and if there is any redundant birth–death point, its point size will be multiplied by the frequency of that point. The example of persistent barcode is shown in Fig. 4(b).

However, the first two illustrations in Fig. 4 are not befitting to create input vectors of k-dimensional topological features for further processing using machine learning method because the number of components of persistent barcode is found to be not a constant and the persistent diagram is a highly sparse image [62]. To overcome this problem, another illustration was used in [48,62] called Betti sequences – a plot of kth persistent Betti numbers over the evolution parameters of the filtration of Rips complexes – as shown in example of Fig. 4(d) and (e) for Betti sequences of connected components and hole respectively. In this study, the same Betti sequences are used to find possible patterns of all time windows $TW^{60}(j)$ and to obtain a new input vectors of k-dimensional topological features for further classification using machine learning method to predict next day direction of KLCI movement. We point out [58,63–65] to interested readers to further explore and comprehend the development of theories and methods behind PH. Nowadays, numerous free software developed to implement PH computation, see [36] for details on such software. In this study, we specifically use R-Package 'TDA' [66] to perform computation on the filtration of the Rips complexes

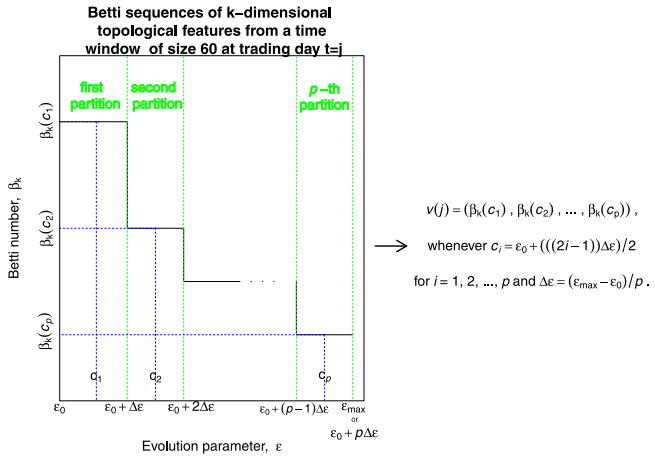

Fig. 5. This figure illustrates on the steps to obtain a $1 \times p$ vector $v\,(j)$ from a Betti sequences of k-dimensional topological features of a time window of size 60 at trading day $t = j$.

built on top of $TW^{60}(j)$ to obtain all persistent Betti numbers for the observed k-dimensional topological features.

To practically combine machine learning methods such as LR, ANN, SVM and RF with PH, we create new input vectors of the observed k-dimensional topological features from the collection of 4612 time windows of size 60 denoted as $\left\{TW^{60}\,(j) \subset X | j = 60, 61, \ldots, 4661\right\}$. Firstly, we apply PH on each time window of size 60 at trading day $t = j$ that is $TW^{60}\,(j)$ to obtain the corresponding Betti sequences plot. From the corresponding Betti sequences plot, we divide equally the plot's horizontal axis given by values of the evolution parameters $\varepsilon \in [\varepsilon_0, \varepsilon_{max}]$ into p partitions of equal size. Herein, partition number p indicates how many scales will be computed from the Betti sequences plot. The p computed scales are the values of kth persistent Betti numbers on the vertical axis of the plot at the middle values of the evolution parameter ε in each partition. Let p and $\Delta\varepsilon = (\varepsilon_{max} - \varepsilon_0)\,/p$ be the partition number and the width of each partition respectively, for each Betti sequences plot, we will obtain a sequence of kth persistent Betti numbers denoted as $\beta_k\,(c_1)\,, \beta_k\,(c_2)\,, \ldots, \beta_k\,(c_p)$, where $c_i = \varepsilon_0 + ((2i-1)\Delta\varepsilon)\,/2$ is the middle value of the evolution parameter ε in each partition for all $i = 1, 2, \ldots, p$. This sequence obtained from the corresponding Betti sequences plot of $TW^{60}(j)$ can be encoded as a $1 \times p$ vector such that $v\,(j) = \left(\beta_k\,(c_1)\,, \beta_k\,(c_2)\,, \ldots, \beta_k\,(c_p)\right)$ whenever $c_i = \varepsilon_0 + ((2i-1)\Delta\varepsilon)\,/2$ for all $i = 1, 2, \ldots, p$ and $\Delta\varepsilon = (\varepsilon_{max} - \varepsilon_0)\,/p$. Please refer to Fig. 5 to enhance your understanding on how to get this $1 \times p$ vector from a Betti sequences plot of $TW^{60}(j)$. The same procedure is applied for the rest of the time windows of size 60. Therefore, for the collection of the 4612 corresponding Betti sequences of each $TW^{60}(j)$ for $j = 60, 61, \ldots, 4661$, we will obtain new input vectors of k-dimensional topological features, denoted as a $4612 \times p$ matrix such that $\left[\begin{array}{cccc} v\,(60) & v\,(61) & \cdots & v\,(4661) \end{array}\right]^T$, where the vector $v\,(j) \in \mathbb{R}^p$ for $j = 60, 61, \ldots, 4661$. Eventually, we can use the input vectors of k-dimensional topological features obtained for classification task using machine learning methods: LR, ANN, SVM and RF.

At this stage, we recognize that there are two tuning parameters involved to create input vectors of k-dimensional topological features: window size (w) and partition number (p). In this study, we vary these pair tuning parameters as follows: $w \in \{10, 30, 60, 100\}$ and $p \in \{30, 60, 100\}$. As a result, 12 input vectors will be created for each of the observed k-dimensional topological features. In particular, only the first two k-dimensional topological features are examined namely the connected components (0-dimensional topological features) and holes (1-dimensional topological features). Therefore, we consider a total of 24 input vectors in this case, comprising 12 input vectors of connected components and 12 input vectors of holes respectively. The aforementioned input vectors are new alternatives to stock return (mentioned in Section 3.1) and also to technical indicators (mentioned in Section 3.2) for the machine learning methods – LR, ANN, SVM and RF – to learn and predict next day direction of KLCI movement. In overall, 26 input vectors are considered since stock return and technical indicators are also examined in this study. In the next subsection, we will introduce the proposed machine learning methods – LR, ANN, SVM and RF – to perform classification task on all these input vectors.

3.5. Machine learning methods

In this section we describe the four machine learning methods proposed in this study namely logistic regression (LR), artificial neural network (ANN), support vector machine (SVM) and random forest (RF). The first three methods are single base models and the last method is ensemble or multiple base model.

3.5.1. Logistic regression

Logistic regression (LR) is the simplest classification model and commonly employed to predict next day direction of stock price movement [10,67,68]. By using LR, the probability (p) for a direction of stock price movement to occur (e.g. increases or decreases) is equal to the logistic sigmoid function, formulated as $p = 1/\left(1 + e^{-f}\right) \in [0, 1]$, where f is a linear function such that$f = b_0 1 + \sum_{i=1}^{n} b_i v\,(i)$, where b_i is coefficient and $v\,(i)$ is the input vectors. The coefficients of the input vectors are computed using a maximum-likelihood technique. In this study, the tuning parameter a which is the chosen cut-off probability value varies in the set of $\{0.10, 0.11, \ldots, 0.90\}$. For classification task, if the $p \geq a$, where $a \in \{0.10, 0.11, \ldots, 0.90\}$, then the direction is classed as increases and vice versa.

3.5.2. Artificial neural network

Artificial neural network (ANN) is among the most common classification models for future direction of stock price movement prediction [11,12,69]. We use a three layer feed-forward ANN optimized by backpropagation and gradient descent approach. We consider an ANN with the input vectors in input layer, one hidden layer of n neurons and two variables of stock price movement direction (increases or decreases) in output layer. Each layer in ANN is connected with each other accordingly, contains a bias parameter (except for output layer) and the neurons of a layer are linked to the neighbor layers with weights. Through the learning process, the initial value for these weights are randomly assigned and the backpropagation and gradient descent approach update the weights to minimize the relative percentage of root mean square. The logistic sigmoid (activation) function is used on the hidden and output layers. As a result, the outputs of the ANN will vary between 0 and 1. For classification, if the output value is equal or greater than 0.5, then the output direction is classified as increase and for otherwise, it is classified as decrease. In this study, tuning parameters in ANN model are varied as followed: the number of neurons in the hidden layer (n) is varied as $n \in \{5, 10, \ldots, 50\}$, the number of iterations (ep) is varied as $ep \in \{100, 200, \ldots, 1000\}$ and the value of learning rate (lr) is fixed to 0.1.

3.5.3. Support vector machine

Support vector machine (SVM) is also another widely applied classification model to predict next day direction of stock price movement [13–15]. In SVM, the main objective is to construct an optimal separating hyperplane (OSH) as separation boundary to distinguish training data (the input vectors) according to their direction class label (increases or decreases). The OSH is obtained by maximizing the margin distance between the separating hyperplane and the training points closest to it (support vectors). Assume a training data as a collection of n pairs, denoted as $\{(v(i), y(i)) | i = 1, 2, \ldots, n\}$, such that $v(i) \in \mathbb{R}^d$ and $y(i) \in \{-1, 1\}$ is the label for input vector $v(i) \in \mathbb{R}^d$, where label -1 is for decrease class and label 1 is for increase class. The optimization problem of SVM to obtain the OSH is given by:

$$\underset{w,\xi}{\text{Minimize}} \left(\frac{1}{2} \|w\|^2 + c \sum_{i=1}^{N} \xi(i) \right) \quad (1)$$

subject to

$$y(i) \left(w^T \Phi(v(i)) + b \right) \geq 1 - \xi(i), (\xi(i) \geq 0, i = 1, 2, \ldots, N). \quad (2)$$

The sum of slack variables, denoted as $\sum_{i=1}^{N} \xi(i)$ represents how much SVM wrongly classify $v(i) \in \mathbb{R}^d$, therefore it is an upper bound on the number of training errors. Here, the cost c in (1) is a positive parameter used to control the bias–variance tradeoff between the number of training errors and the margin distance under the constraint of (2). The $\Phi: \mathbb{R}^d \rightarrow H$ in (2) is a feature map that maps $v(i) \in \mathbb{R}^d$ into a higher dimensional space $\Phi(v(i)) \in H$ where training data becomes linearly separable. Here, the similarity (or dissimilarity) of $\Phi(v(i)) \in H$ can be computed by a positive kernel function defined as $K(v(i), v(j)) = \langle \Phi(v(i)), \Phi(v(j)) \rangle_H$, which is an inner product in the space H. Therefore, if the positive kernel function can be defined, SVM with kernel trick can be performed to obtain the OSH (details on SVM with kernel trick can be read from [52]). In this study, we use the radial basis kernel function defined as $K(v(i), v(j)) = \exp\left(-\gamma \|v(i) - v(j)\|^2\right)$, where gamma γ is the inverse of the standard deviation of the kernel function. Therefore, there are two tuning parameters involved in this SVM model, which are cost c and gamma γ in radial basis kernel function. Those two tuning parameters are important to obtain the OSH and vary as follows: $c \in \left\{2^{-5}, 2^{-4}, \ldots, 2^{10}\right\}$ and $\gamma \in \left\{2^{-50}, 2^{-49}, \ldots, 2^{10}\right\}$.

3.5.4. Random forest

Random forest (RF) is an ensemble of some decision trees developed to obtain better prediction performance as compared to a single decision tree [18,19,70]. Each decision tree of RF is built on a bootstrap sample using binary recursive partitioning (BRP) [71]. In the BRP algorithm, a random subset of the input vectors is initially selected and all possible splits of all the input vectors are then evaluated. Subsequently, the best split obtained is used to create a binary classification. All these process are repeated recursively within each subsequent partition and finished when the partition size equal to 1. There are two tuning parameters involved in modeling RF, those are the number of trees in the ensemble (p) and the number of the input vectors to try at each split (k). In this study, the two parameters being varied are as follows: $p \in \{50, 100, \ldots, 500\}$ and $k \in \{2, 4, \ldots, 20\}$.

3.6. Classification task

In our classification task, a total of 26 input vectors are considered for binary classification task using machine learning methods – LR, ANN, SVM and RF – to learn and predict the next day direction of Kuala Lumpur Composite Index (KLCI) index movement. These input vectors are stock returns, technical indicators, 12 input vectors of connected components and 12 input vectors of holes. For valid comparison, similar steps of binary classification task are applied on all these input vectors. Since binary classification task using the stated machine learning methods require two classes of labeled input vectors, we label each input vector as follows: Let $\{(v(i), y(i)) | i = 1, 2, \ldots, q\}$ denote a collection of labeled input vectors, where the pair $(v(i), y(i))$ for trading day $i = 1, 2, \ldots, q$ is a two classes labeled input vector such that $v(i) \in \mathbb{R}^d$ and $y(i) \in \{-1, 1\}$ be the label for vector $v(i) \in \mathbb{R}^d$, then $y(i) = 1$ at the trading day i if the next day stock return of the KLCI index at the trading day $i + 1$ is equal or larger than zero (for increases direction class) and $y_i = -1$ if otherwise (for decreases direction class). Table 2 provides the total number of increases and decreases direction classes for the next day direction of the KLCI movement in each year from 2001 until 2017. Note that, to maintain data consistency, we exclude data collected from 2000 and 2018 as the total amount of data from year 2000 varies depending on window size w due to daily sliding window approach while data from year 2018 is incomplete. Therefore, we have a total of 17 years of data from 2001 until 2017 with 4434 trading days to be used in the classification task. Here, classification task using those machine learning methods — LR, ANN, SVM and RF — are performed on year —to— year basis and this is similar to what have been done in [11]. As stock price date are constantly changing, it is inevitable to conduct periodic classification task in this manner.

Table 2
The total number of increases and decreases direction classes for the next day direction of the KLCI movement in each year from 2001 until 2017.

Year	Increases	Increases %	Decreases	Decreases %	Total
2001	140	53.64	121	46.36	261
2002	136	52.11	125	47.89	261
2003	151	57.85	110	42.15	261
2004	149	56.87	113	43.13	262
2005	138	53.08	122	46.92	260
2006	166	63.85	94	36.15	260
2007	163	62.45	98	37.55	261
2008	120	45.80	142	54.20	262
2009	154	59.00	107	41.00	261
2010	158	60.54	103	39.46	261
2011	141	54.23	119	45.77	260
2012	160	61.30	101	38.70	261
2013	146	55.94	115	44.06	261
2014	144	55.17	117	44.83	261
2015	133	50.96	128	49.04	261
2016	139	53.26	122	46.74	261
2017	142	54.83	117	45.17	259
Total	2480	55.93	1954	44.07	4434

Before applying any machine learning methods, we normalize values of all input vectors to a common scale [0, 1] by dividing them with the largest maximum value in each feature (column of the input vectors) as applied in [48]. Normalization is crucial to avoid influence of outliers and to equalize scales of invariant topological features (connected components and holes) from the different parameter values of window size w and partition number p. Furthermore, we face imbalanced direction classes for the labeled input vectors over the years, as shown in Table 2. We use under-resampling method to circumvent this problem, where we randomly select a sample subset from these two classes so that their elements match for each year. As an example, for year 2001 in Table 2, we randomly pick 120 out of 140 increases and 121 decreases classes (120 is chosen since it is the closest smaller even integer to 121, i.e. the total element in the smaller class). In total, we have a subset of 240 total elements containing 120 increases and 120 decreases classes that are split into equal-sized classes for training data (50% of the total) and testing data (50%

of the total). Table 3 shows the balanced equal-sized classes of training and testing data for 2001 until 2017. Data balancing is applied to avoid overfitting of the major class during learning of the machine learning methods. Training data is created for learning process of the machine learning methods and to find optimal values for their tuning parameters, while testing data is used to compute the prediction performance.

Here, for each training data from the observed years in Table 3, we perform grid search approach in finding optimal tuning parameters involved in modeling machine learning methods – LR, ANN, SVM and RF – with PH. Table 4 shows all the tuning parameters and their grid levels for PH and these machine learning methods respectively. For cases without PH, PH's tuning parameters are not involved and the machine learning methods are independently applied to stock returns and also to technical indicators. For each training data, by using grid search approach, we compute each prediction performance covering all possible combination of the grid levels. Let P be the prediction performance on training (or testing) data consisting of m vectors, then P is computed using classification accuracy score (in percentage) formulated as follows:

$$P = \left(\frac{1}{m}\sum_{i=1}^{m} R(i)\right) \times 100\% \text{ for } i = 1, 2, \ldots, m,$$

where $R(i)$ is the prediction result at the trading day i is defined as follows:

$$R(i) = \begin{cases} 1 & \text{if } PO(i) = AO(i) \\ 0 & \text{otherwise,} \end{cases}$$

where $PO(i)$ and $AO(i)$ are the prediction direction output at the trading day i from any machine learning method and actual direction output at the trading day i respectively. In this study, we choose the optimal tuning parameters for the machine learning methods with PH (or without PH) based on the highest prediction performance given on the training data. Next, the machine learning methods – LR, ANN, SVM and RF – with PH (or without PH) along with their optimal tuning parameters are used to predict testing data from the observed years in Table 3 in order to analyze prediction performance in predicting the next day direction of KLCI index movement.

3.7. Evaluation measures

To evaluate which combination of the machine learning methods – LR, ANN, SVM and RF – with PH (or without PH) provide the best prediction performance, we use the average of the prediction performances Ps on testing data (as described in 3.6) and pairwise model comparison method to compare prediction performances for the respective machine learning methods on four different input vectors, which are stock returns, technical indicators, input vectors of connected components and input vector of holes. By using the average, denoted as $\left(\sum_{i=1}^{17} P_i\right)/17$, machine learning method with input vectors that provide the highest average of these Ps on testing data is said to be the best combination method to predict the direction of KLCI movement.

For pairwise model comparison method, we construct a pairwise table with combination of each machine learning method (LR, ANN, SVM and RF) with each input vectors (stock returns, stock returns, technical indicators, input vectors of connected components and input vectors of holes) are listed along the top row and left-hand side of the table respectively. Next, we place dashes diagonally downwards in the table. For $i \neq j$, let C_i and C_j are two different combination methods, we fill in the whole table by comparing two values based on P on testing data obtained from these C_i and C_j respectively to determine which method combination generates better performance. This comparison process is repeated for all years. For each observed year, we add 1, 0.5 or 0 at each time in the table of row i and column j if C_i has higher value of P on testing data than C_j, if these two C_i and C_j have a similar value of P on testing data and if C_i has lower value of P on testing data than C_j respectively. Afterwards, we sum across each row to determine the total scores. In this study, we also conclude that the combined method stated in left-hand side of the table with the highest value in the total score column is the finer combination method to predict the direction of KLCI movement.

4. Result

This section presents all empirical results obtained from the proposed methods above. Firstly, we provide empirical results for the machine learning methods – logistic regression (LR), artificial neural network (ANN), support vector machine (SVM) and random forest (RF) – independently on stock returns of Kuala Lumpur Composite Index (KLCI), Kuala Lumpur Stock Exchange Industrial (KLSE IND) and Kuala Lumpur Stock Exchange Technology (KLSE TEC) indexes. Next, we provide empirical results for these machine learning methods which are independent on technical indicators (Simple 10-days moving average, Weighted 10-days moving, Momentum, Stochastic K%, Stochastic D%, Relative strength index and Moving average converge diverge). Then, we illustrate plots of the normalized Betti sequences of connected components and normalized Betti sequences of holes obtained from the stock returns using persistent homology (PH) to observe their patterns for increases and decreases direction classes. Subsequently, we provide empirical results from the application of machine learning methods on input vectors of connected components and also application of machine learning methods on input vectors of holes. By using classification accuracy score, prediction performances obtained from testing data of machine learning methods on four different input vectors (namely stock returns, technical indicators, input vectors of connected components and input vectors of holes) are computed and summarized in the following figures.

4.1. Empirical results for machine learning methods on stock returns

In the first part of our empirical study, we apply selected machine learning methods independently on stock returns of the KLCI, KLSE IND and KLSE TEC indexes with the objective to predict the next day direction of KLCI movement. Empirical results for each year from 2001 to 2017 are summarized in Fig. 6. Based on classification accuracy score on testing data, the results in Fig. 6(a) shows that the prediction performances for LR vary from 50% to 63.27% with an average of 55.32%. In Fig. 6(b), prediction performances for ANN vary from 52.73% to 63.27% with an average of 57.81%. Furthermore, prediction performances for SVM vary from 53.23% to 64.29% with an average of 58.25% as shown in Fig. 6(c). In the case of RF, Fig. 6(d) shows prediction performances vary from 47.41% to 63.27% with an average of 52.90%.

4.2. Empirical results for machine learning methods on technical indicators

For the second empirical study, we apply selected machine learning methods independently on technical indicators (Simple 10-days moving average, Weighted 10-days moving, Momentum, Stochastic K%, Stochastic D%, Relative strength index and Moving average converge diverge) to predict the next day direction of KLCI movement. Empirical results for each year from 2001 to 2017 are summarized in Fig. 7. Based on classification accuracy

Table 3
The balanced equal-sized classes of training and testing data for 2001 until 2017.

Year	Equal-sized classes training data (50% of input data)			Equal-sized classes testing data (50% of input data)		
	Increases	Decreases	Total	Increases	Decreases	Total
2001	60	60	120	60	60	120
2002	62	62	124	62	62	124
2003	55	55	110	55	55	110
2004	56	56	112	56	56	112
2005	61	61	122	61	61	122
2006	47	47	94	47	47	94
2007	49	49	98	49	49	98
2008	60	60	120	60	60	120
2009	53	53	106	53	53	106
2010	51	51	102	51	51	102
2011	59	59	118	59	59	118
2012	50	50	100	50	50	100
2013	57	57	114	57	57	114
2014	58	58	116	58	58	116
2015	64	64	128	64	64	128
2016	61	61	122	61	61	122
2017	58	58	116	58	58	116

Table 4
The corresponding tuning parameters and their grid levels based on PH and other machine learning methods.

Method	Tuning parameter	Grid levels
Persistent homology (PH)	Window size (w) Partition number (p)	10, 30, 60, 100 30, 60, 100
Logistic regression (LR)	Cut-off probability (a)	0.10, 0.11, . . . , 0.90
Artificial neural network (ANN)	Number of neurons in the hidden layer (n) Number of iterations (ep) Learning rate (lr)	5, 10, . . . , 50 100, 200, . . . , 1000 0.1
Support vector machine (SVM)	Cost (c) Gamma (γ)	$2^{-5}, 2^{-4}, \ldots, 2^{10}$ $2^{-50}, 2^{-49}, \ldots, 2^{10}$
Random forest (RF)	Number of trees (p) Number of input vectors for each split (k)	50, 100, . . . , 500 2, 4, . . . , 20

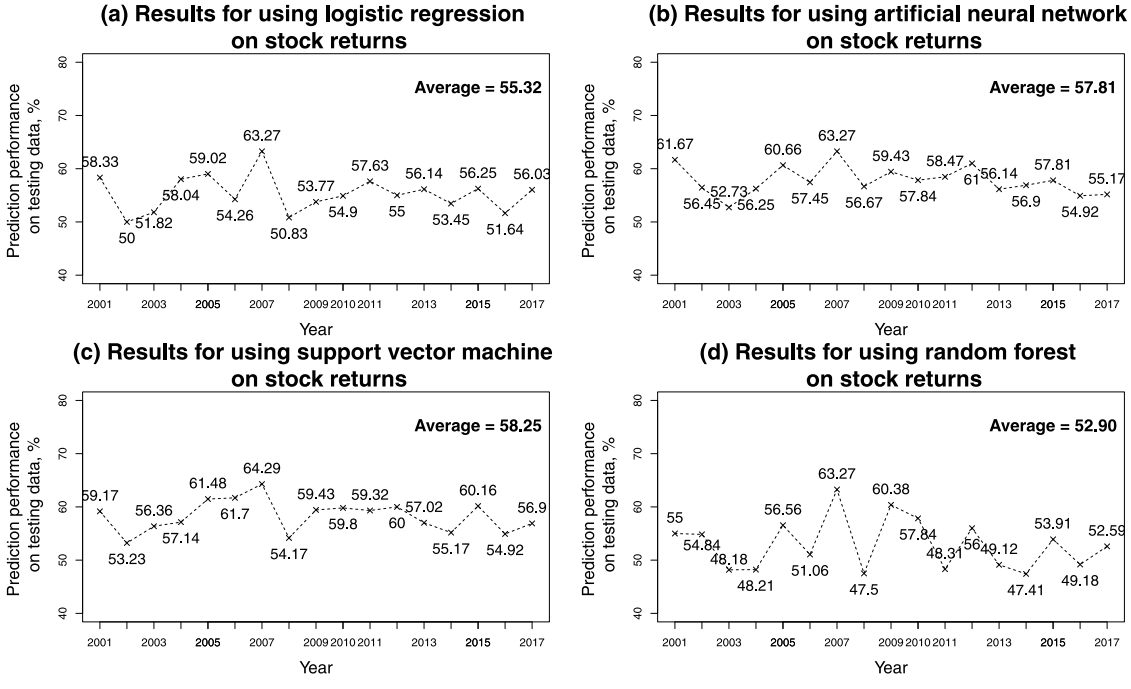

Fig. 6. Empirical results using LR, ANN, SVM and RF on stock returns. See appendices A1–A4 for details of these results.

score on testing data, the results in Fig. 7(a) show that prediction performances for LR vary from 53.28% to 66.67% with an average of 59.17%. In Fig. 7(b), prediction performances for ANN vary from 53.91% to 70% with an average of 60.97%. Furthermore, prediction performances for SVM vary from 56.90% to 75.51% with an average of 64.21% as shown in Fig. 7(c). In the case of RF, Fig. 7(d) shows that prediction performances vary from 49.18% to 71.43% with an average of 60.18%.

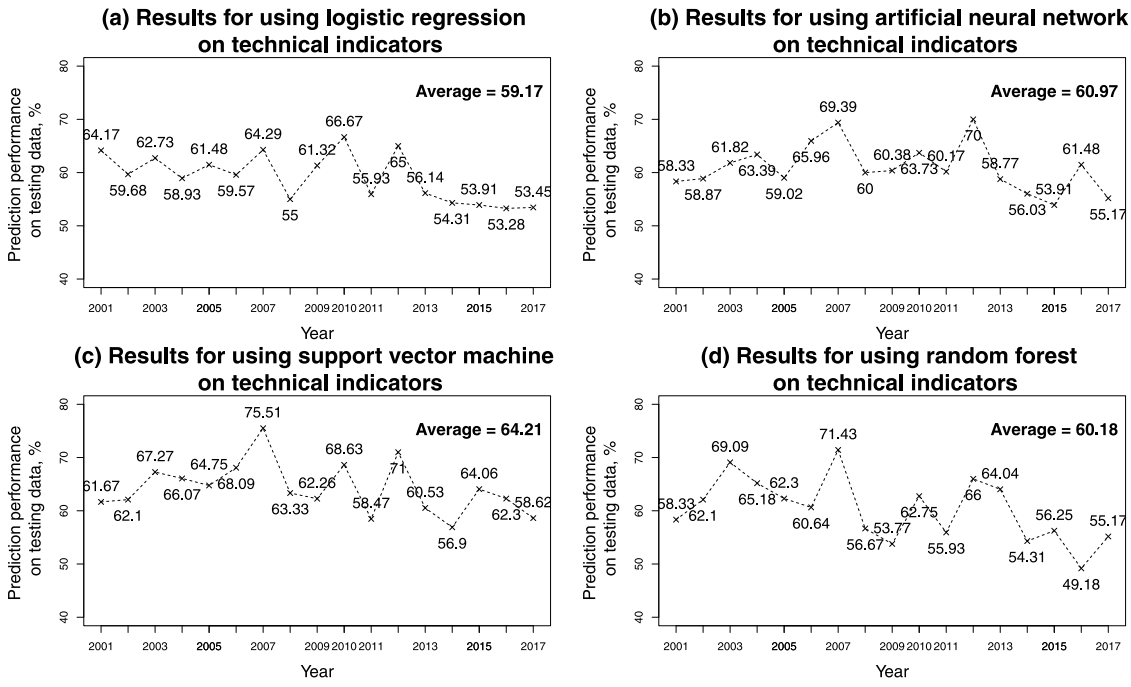

Fig. 7. Empirical results when using LR, ANN, SVM and RF on technical indicators. See appendices A5–A8 for details of these results.

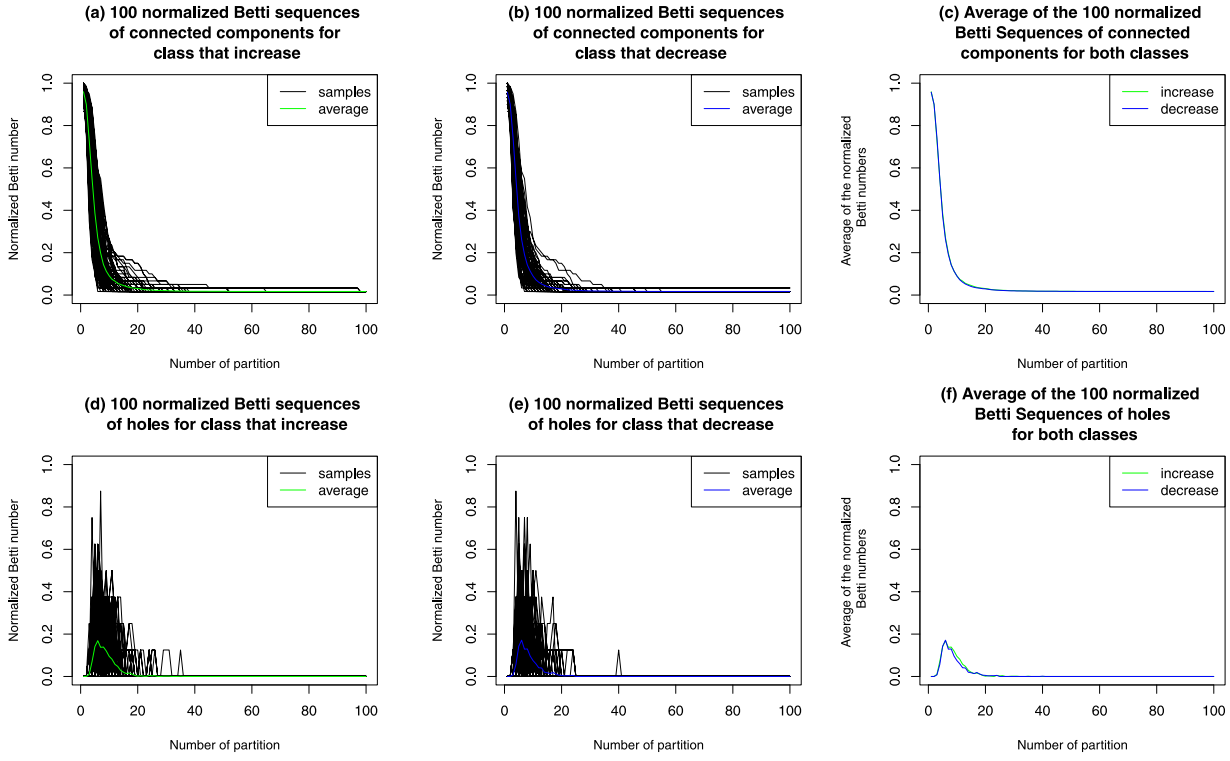

Fig. 8. In the first row, 100 sample plots of the normalized Betti sequences of connected components for increases class, 100 sample plots of the normalized Betti sequences of connected components for decreases class and the average of the plots for those two classes are illustrated. In the second row, 100 sample plots of the normalized Betti sequences of holes for increases class, 100 sample plots of the normalized Betti sequences of holes for decreases class and the average of the plots for these two classes are also shown.

4.3. Plots of the normalized Betti sequences

In this sequel, we take the initiative to extend our study by using PH to obtain plots of the normalized Betti sequences of connected components and also the normalized Betti sequences of holes from the stock returns. These two plots are illustrated to observe their patterns for increases and decreases classes for the next day direction of KLCI movement. Here, the normalized Betti sequences of the topological features (connected components and holes) is a plot of the normalized persistent Betti numbers of the topological features versus certain evolution parameters of the filtration of Rips complexes built on top of a time window of

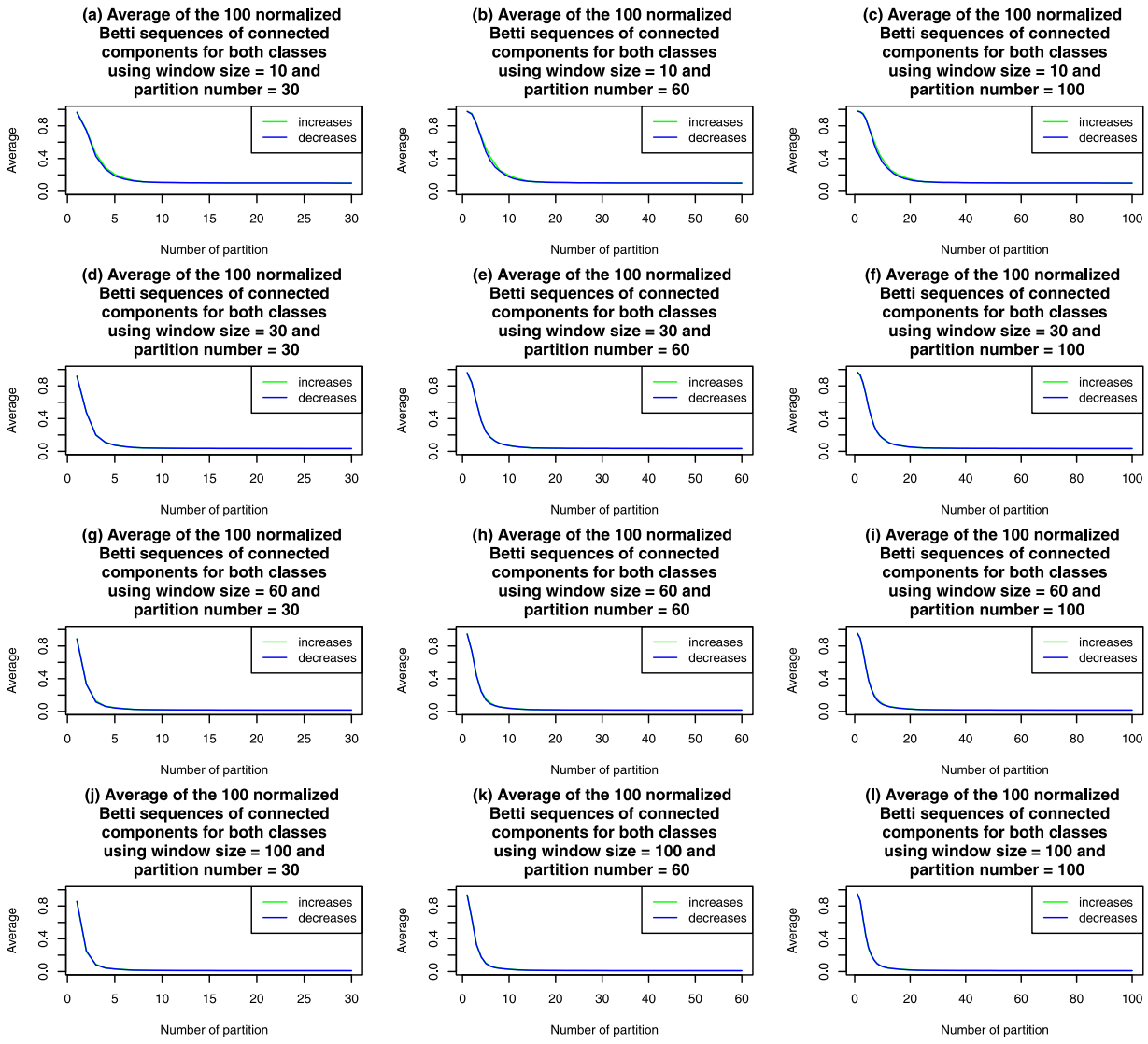

Fig. 9. Averages of the 100 normalized Betti sequences of connected components for increase and decrease classes with different pair values of the two tuning parameters: window size $w \in \{10, 30, 60, 100\}$ and partition number $p \in \{30, 60, 100\}$ respectively.

stock returns. Fig. 8 illustrates 100 sample plots of the normalized Betti sequences of connected components and 100 sample plots of the normalized Betti sequences of holes that are randomly selected to represent increases and decreases classes respectively. The normalized Betti sequences in Fig. 8 are obtained using value parameters of window size $w = 60$ and partition number $p = 100$. The average of the 100 normalized Betti sequences of connected components and the average of the 100 normalized Betti sequences of holes are shown in Fig. 8 accordingly.

Further, we also vary the values for tuning parameters of window size w and partition number p. The average of the 100 normalized Betti sequences of connected components and the average of the 100 normalized Betti sequences of holes with respect to different pair values of those tuning parameters of window size, $w \in \{10, 30, 60, 100\}$ and partition number, $p \in \{30, 60, 100\}$ are presented in Figs. 9 and 10 respectively.

4.4. Empirical result for SVM with PH

In this subsection, we provide our main empirical results from the application of selected machine learning methods – LR, ANN, SVM and RF – on input vectors of connected components and also on input vectors of holes. Detailed results of prediction performances on testing data from year 2001 to 2017 for input vectors of connected components and input vectors of holes are provided in Figs. 11 and 12 respectively.

When using input vectors of connected components obtained from PH as features for the proposed machine learning methods, Fig. 11(a) shows that prediction performances of using LR on testing data for each year from 2001 to 2017 vary from 59.02% to 75.51% with an average of 64.84%. Meanwhile, in Fig. 11(b), ANN's prediction performances on testing data vary from 58.47% to 73.47% with an average of 64.89%. In Fig. 11(c), SVM's prediction performances on testing data vary from 61.02% to 76.53% with an average of 67.15%. Moreover, in Fig. 11(d), RF's prediction performances on testing data for the similar years vary from 60.17% to 77.55% with an average of 66.61%.

Input vector of holes obtained from PH is also considered as another feature to be further classified by the proposed machine learning methods. The result of using LR in Fig. 12(a) shows that prediction performances on testing data for each year from 2001 to 2017 vary from 56.78% to 74.49% with an average of 63.65%. Likewise, prediction performances for ANN on testing data for similar years vary from 56.78% to 76.53% with an average of 65.87%, as shown in Fig. 12(b). For SVM, Fig. 12(c) shows that prediction performances on testing data from 2001 until 2017

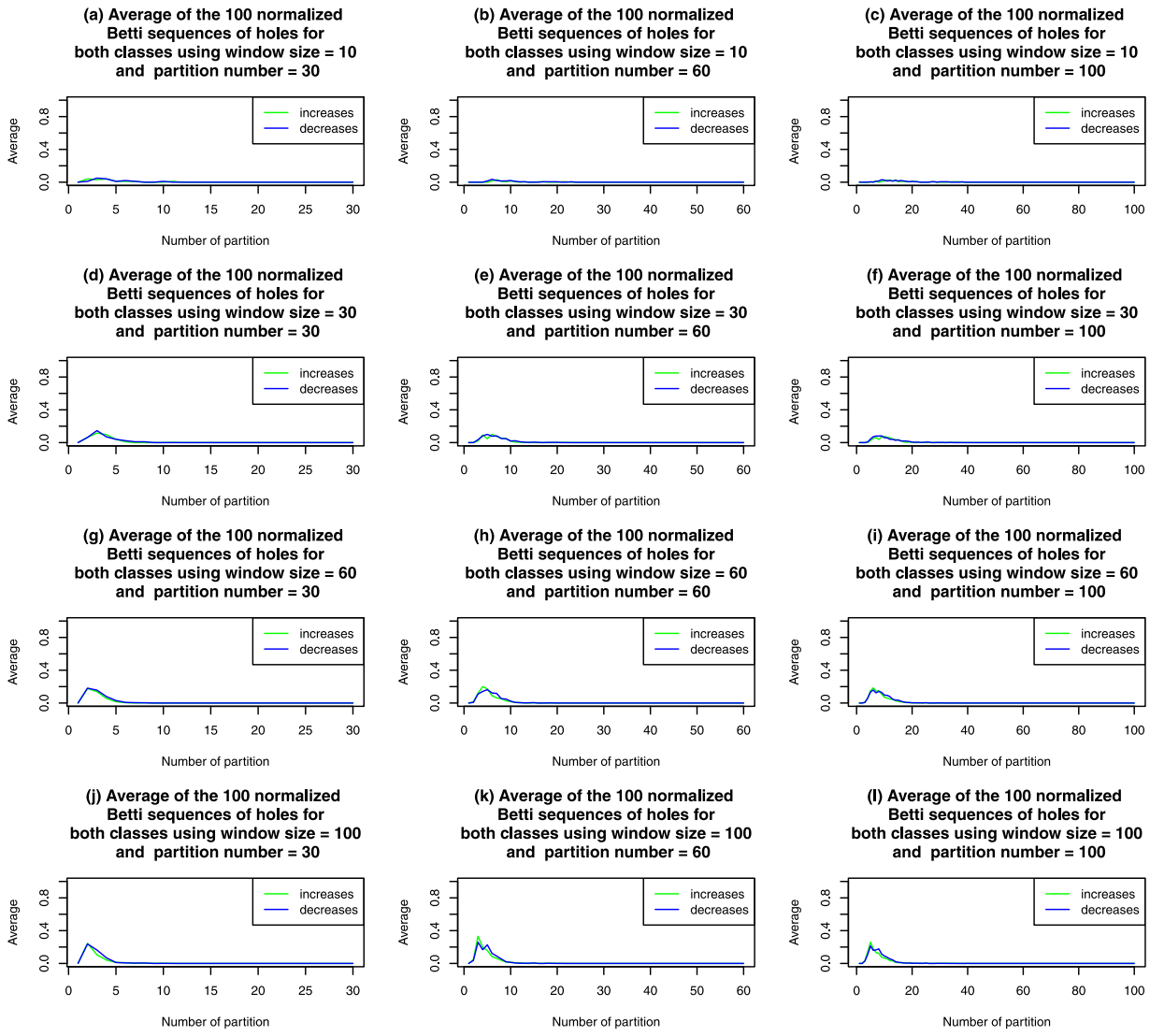

Fig. 10. Averages of 100 normalized Betti sequences of holes for increases and decreases classes with different pair values of the two tuning parameters: $w \in \{10, 30, 60, 100\}$ and $p \in \{30, 60, 100\}$ respectively.

vary from 57.63% to 76.53% with an average of 66.68%. Moreover, prediction performances on testing data for RF in Fig. 12(d) vary from 56.78% to 76.53% with an average of 65.74%.

4.5. Pairwise model comparison method

Besides the average of prediction performances, we also utilize the pairwise model comparison method (as described in Section 3.7) to evaluate which combination of machine learning methods (LR, ANN, SVM and RF) with input vectors (stock returns, technical indicators, connected components and holes) provides the best prediction performance. Complete results generated by using the pairwise model comparison method are provided in appendix A17. In Table 5, we provide the summarized results from the average of prediction performances and the total score of pairwise model comparison method respectively.

5. Discussion of the results

In this study, we use two evaluation methods namely the average of prediction performances and the total score of pairwise model comparison method to compare empirical results when applying those machine learning methods – logistic regression (LR), artificial neural network (ANN), support vector machine (SVM) and random forest (RF) – on stock returns, applying those machine learning methods on technical indicators, applying those machine learning methods on input vectors of connected components and applying those machine learning methods on input vectors of holes to determine which method combination produced the best prediction performance. Next, we elaborate the efficiency of PH to provide input vectors of topological features (connected components and holes) for further processing with those machine learning methods in predicting the next day direction of Kuala Lumpur Composite Index (KLCI) movement.

5.1. Discussion of machine learning methods on stock returns

From Section 4.1, the prediction performances on testing data obtained confirm that the probability of each of the proposed machine learning method on stock returns to accurately predict the next day direction is above 50% (better than flipping a coin), except for RF (see year 2003 and 2014 as examples). It presents some limitations of RF to attain good prediction performances, making this method unreliable relative to other methods. Based on the results in Table 5, superior results are seen for SVM with

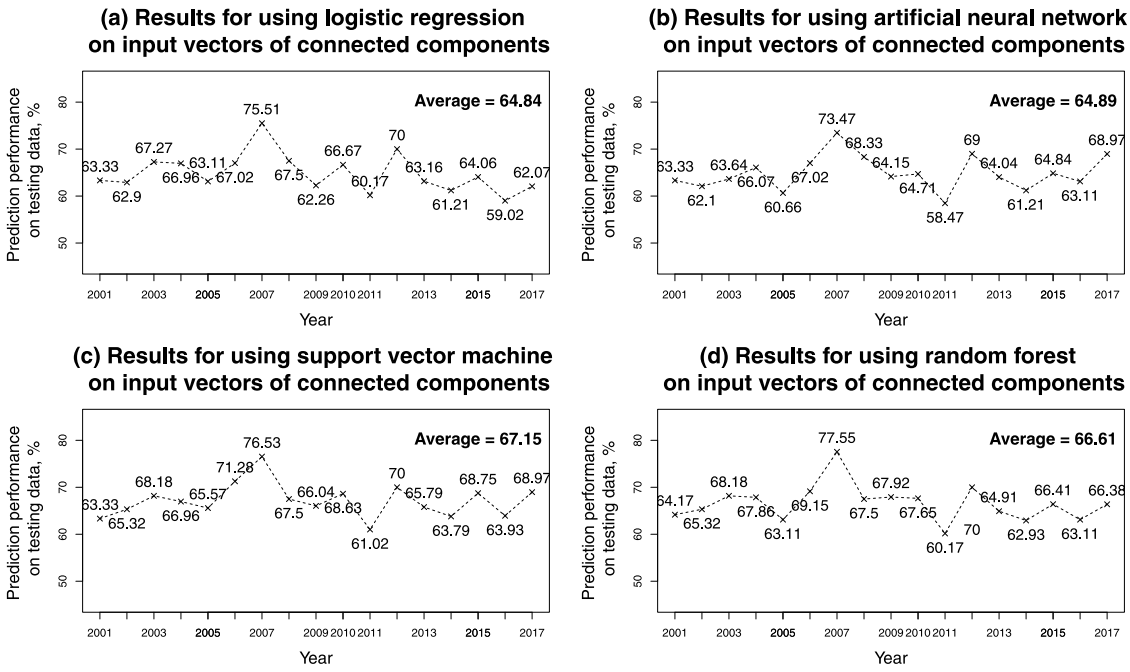

Fig. 11. Empirical results when using LR, ANN, SVM and RF on input vectors of connected components. See appendices A9–A12 for details of these results.

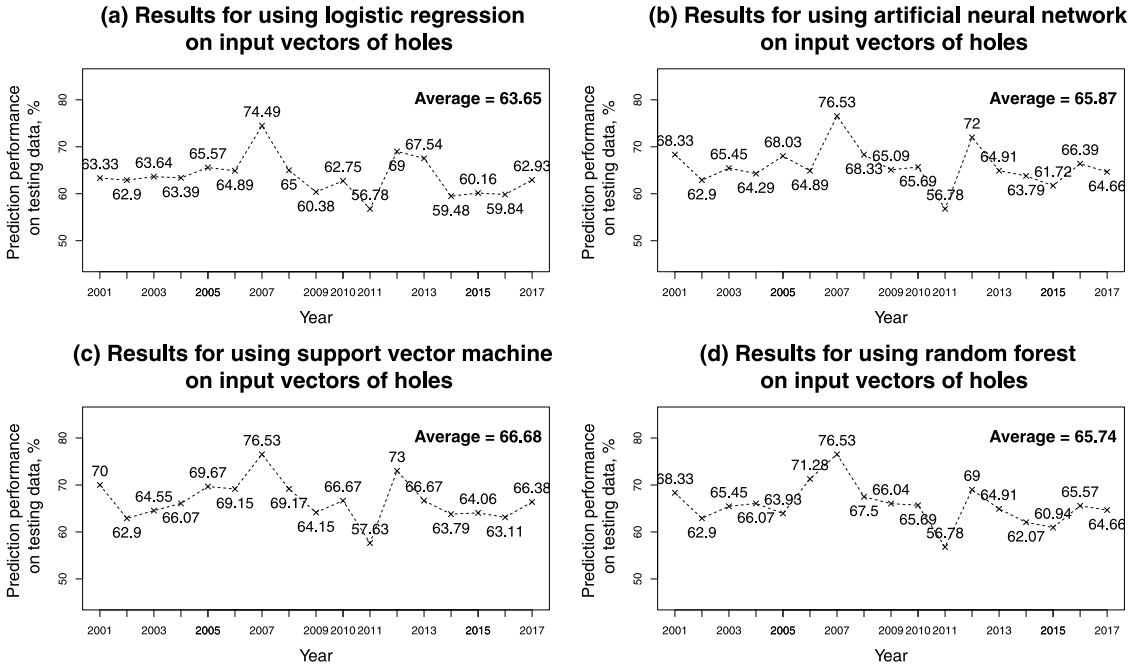

Fig. 12. Empirical results when using LR, ANN, SVM and RF on input vectors of holes. See appendices A13–A16 for details of these results.

performance average of 58.25%, surpasses other machine learning methods such as LR, ANN and RF with performance averages of 55.32%, 57.81% and 52.90% respectively. For total score of pairwise model comparison method, the results lead to similar conclusion where SVM with the total score of 69.50 surpasses other machine learning methods such as LR, ANN and RF with the total score of 30.5, 57 and 10 respectively.

5.2. Discussion of machine learning methods on technical indicators

As discussed in Section 4.2, a similar pattern of results was obtained as per results of Section 4.1 when the prediction performances on testing data produced reveals that the probability of each of the proposed machine learning method on technical indicators to accurately predict the next day direction is above 53.28% (better than flipping a coin), except a year of 2016 for RF. From the results summarized in Table 5, it is clear that SVM with performance average of 64.21% outperforms other machine learning methods such as LR, ANN and RF with performance averages of 59.17%, 60.97% and 60.18% respectively. A similar conclusion was reached by using total score of pairwise model comparison method whereby SVM with the total score of 156 outperforms other machine learning methods such as LR, ANN and RF with the total score of 72, 90 and 83 respectively.

Table 5
The summarized results from the average of prediction performances and the total score of pairwise model comparison method.

No.	Input vectors	Machine learning method	Average of prediction performances (%)	Total score of pairwise model comparison method
1	Stock returns	LR	55.32	30.5
2	Stock returns	ANN	57.81	57
3	Stock returns	SVM	58.25	69.50
4	Stock returns	RF	52.90	10
5	Technical indicators	LR	59.17	72
6	Technical indicators	ANN	60.97	90
7	Technical indicators	SVM	64.21	156
8	Technical indicators	RF	60.18	83
9	Connected components	LR	64.84	163
10	Connected components	ANN	64.89	155
11	Connected components	SVM	67.15	220.5
12	Connected components	RF	66.61	209.5
13	Holes	LR	63.65	132
14	Holes	ANN	65.87	185.5
15	Holes	SVM	66.68	204
16	Holes	RF	65.74	179.5

5.3. Discussion for plots of the normalized Betti sequences

From the observation of figures illustrated in Section 4.3, we obtain the following insights on Betti sequences. In the first row of Fig. 8, the comparison of plots of the normalized Betti sequences of connected components for both increases (in the left box) and decreases (in the middle box) classes reflect similar pattern in logarithmic function shape and their averages (in the right box) which are almost overlapped. It presents limitation to distinguish between these two classes using plots of the normalized Betti sequences of connected components. Meanwhile, the comparison of plots of the normalized Betti sequences of holes for both increases (in the left box) and decreases (in the middle box) classes in second row of Fig. 8 does not provide any significant difference for these two classes when both have similar pattern of tent shaped function. Besides, the average of the normalized Betti sequences of holes for increases class is only slightly higher than decreases class. Therefore, the limitation become apparent to visually distinguish between these two classes based on plots of the normalized Betti sequences of holes.

All of the above mentioned behaviors of the average of 100 normalized Betti sequences of connected components and the average of 100 normalized Betti sequences of holes remain unchanged from alteration values for tuning parameters of window size w and partition number p. Based on our observation and comparison on the averages in Figs. 9 and 10 accordingly, we can conclude that the challenge to obtain a rule-based classification based on patterns of plots of the normalized Betti sequences of connected components and patterns of plots of the normalized Betti sequences of holes, remains. However, machine learning tools such as LR, ANN, SVM and RF can learn to distinguish input vectors of topological features (connected components and holes) that are difficult to be differentiated by human eyes [46–48]. Therefore, our results when using PH cast a new light to obtain new alternative input vectors of connected components and input vectors of holes for further processing using the selected machine learning methods to predict the next day direction of KLCI index movement, as proposed in Section 3.

5.4. Discussion for machine learning methods with PH

As presented in Section 4.4, we observed that all the prediction performances obtained on testing data for the proposed machine learning methods with PH in predicting the next day direction are above 56% and these prediction performances clearly gives better results than those discussed in Sections 5.1 and 5.2 respectively. Referring to Table 5, for the first case of using input vectors of connected components as features for machine learning methods, prediction performances obtained signify that SVM once again outperform other machine learning methods with performance average of 67.15% while LR, ANN and RF obtained performance averages of 64.84%, 64.89% and 66.61% respectively. This result is in line with the total score of pairwise model comparison method when SVM outperform other machine learning methods with the total score of 220.5 while LR, ANN and RF obtained the total score of 163, 155 and 209.5 respectively. In the second case of using input vectors of holes as features for machine learning methods, the analysis also confirmed the above findings in which SVM emerged to be the finest method for the prediction by obtaining the highest performance average of 66.68% as compared to other methods namely LR (63.65%), ANN (65.87%) and RF (65.74%). Invariable, total score of pairwise model comparison method for the second case also provide evidence that SVM is the finest method for the prediction by scoring the highest total score of 204 as compared to other methods namely LR (132), ANN (185.5) and RF (179.5).

5.5. Discussion of the overall results

The present study confirmed the findings that SVM is the finer machine learning method to predict the next day direction of KLCI movement, as compared to LR, ANN and RF. These results also demonstrate that SVM with PH (connected component and holes) provides better result for predicting the next day direction of KLCI movement, as compared to using SVM independently on stock returns and also on technical indicators. In particular, the best average prediction performance and the highest total score of pairwise model comparison method in the overall study are obtained through SVM on input vectors of connected components. Table 6 summarizes the overall results of this study which found clear support on the ability of PH to provide robust and useful topological features as an emerging alternative input vectors to be used with SVM and also other machine learning methods such as LR, ANN and RF to improve prediction performance and this combination is a promising tool in predicting the next day direction of KLCI movement.

In the domain of predicting direction of stock price movement, our result obtained the highest prediction performance of 67.15% and this is consistent and comparable with other prediction results in the literature [11,22,67] and [35]. In [67], this study uses several machine learning methods and obtained the highest average prediction performances of 66.67%. By using SVM with radial basis function, [11] obtained average of prediction performance of 64.00%. In [22], this study perform machine learning methods on so called financial network indicators for next week, next 4

Table 6
The overall empirical results obtained in this study.

Input vectors	The best prediction method based on input vectors	Average of prediction performances (%)	Total score of pairwise model comparison method
Stock returns	SVM	58.25	69.50
Technical indicators	SVM	64.21	156
Connected components	SVM	67.15	220.5
Holes	SVM	66.68	204

weeks, next 8 weeks and next 12 weeks prediction and demonstrated that the prediction accuracy increases correspond with the increase of those weeks. Furthermore, their result shows that for next week prediction (the closest case to our next day prediction) obtained average of prediction accuracy 59.30%. In [35], this study also investigate several machine learning methods and obtained the highest prediction accuracy 62.50%. Furthermore, the exploration of applying PH in predicting direction of stock price movement of our study is still in early state and other variants of persistent Betti numbers (e.g. persistence landscape, persistence image, etc.) may also provide promising results in learning task [72].

6. Conclusion and future works

This study has explored the combination of the commonly used machine learning methods – logistic regression (LR), artificial neural network (ANN), support vector machine (SVM) and random forest (RF) – with persistent homology (PH) as a potential tool for predicting the next day direction of stock price movement. In particular, PH is introduced to improve prediction performance when machine learning methods are used independently on stock returns and also on technical indicators to predict the next day direction of Kuala Lumpur Composite Index (KLCI) movement. By using PH, new input vectors of invariant topological features are obtained from stock returns for further processing using the machine learning methods. The observed invariant topological features used in this study are connected components and holes.

By using the average of prediction performances and the total score of pairwise model comparison method, our empirical results demonstrated that prediction performance on the dataset using the proposed machine learning methods – LR, ANN, SVM and RF – on input vectors of invariant topological features (connected components or holes) is better than using the machine learning methods independently on stock returns and also on technical indicators. Furthermore, SVM on input vectors of connected components produced the best prediction performance compared to other methods. In general, this implies that the combination of machine learning methods with PH offers promising alternative tool in predicting the next day direction of stock price movement.

There are other research recommendations that may enhance prediction performance using machine learning methods together with PH. In future research, the prediction performances may be improved with a more refined feature selection method applied to input vectors of connected components and input vectors of holes in obtaining their most relevant features. On another note, interesting future research could include exploring PH on all stock price's components such as opening price, highest price, lowest price, closing price and stock volume. In addition, as mentioned by [72], other variants of persistent Betti numbers (e.g. persistence landscape, persistence image, etc.) also provide promising results in learning task. Therefore, future research should further develop using machine learning method with PH which carries a huge potential to generate a finer result.

CRediT authorship contribution statement

Mohd Sabri Ismail: Conceptualization, Methodology, Software, Validation, Writing - original draft, Writing - review & editing. **Mohd Salmi Md Noorani:** Conceptualization, Supervision, Validation, Writing - review & editing, Funding acquisition. **Munira Ismail:** Supervision, Data curation, Validation, Writing - review & editing. **Fatimah Abdul Razak:** Supervision, Validation. **Mohd Almie Alias:** Supervision, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank Universiti Kebangsaan Malaysia and Centre for Research and Instrumentation (CRIM) for the financial funding through FRGS/1/2019/STG06/UKM/01/3.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.asoc.2020.106422.

References

[1] M. Göçken, M. Özçalıcı, A. Boru, A.T. Dosdoğru, Integrating metaheuristics and artificial neural networks for improved stock price prediction, Expert Syst. Appl. 44 (2016) 320–331.
[2] R.C. Cavalcante, R.C. Brasileiro, V.L.F. Souza, J.P. Nobrega, A.L.I. Oliveira, Computational intelligence and financial markets: A survey and future directions, Expert Syst. Appl. 55 (2016) 194–211.
[3] F. Zhou, Q. Zhang, D. Sornette, L. Jiang, Cascading logistic regression onto gradient boosted decision trees for forecasting and trading stock indices, Appl. Soft Comput. 84 (2019) 105747.
[4] F.E.H. Tay, L. Cao, Application of support vector machines in financial time series forecasting, Omega 29 (2001) 309–317.
[5] N. Zhang, A. Lin, P. Shang, Multidimensional k-nearest neighbor model based on EEMD for financial time series forecasting, Physica A 477 (2017) 161–173.
[6] P.C.S. Bezerra, P.H.M. Albuquerque, Volatility forecasting via SVR–GARCH with mixture of Gaussian kernels, Comput. Manag. Sci. 14 (2017) 179–196.
[7] X. Zhong, D. Enke, Forecasting daily stock market return using dimensionality reduction, Expert Syst. Appl. 67 (2017) 126–139.
[8] H. Chen, K. Xiao, J. Sun, S. Wu, A double-layer neural network framework for high-frequency forecasting, ACM Trans. Manag. Inf. Syst. 7 (2017) 11.
[9] B.M. Henrique, V.A. Sobreiro, H. Kimura, Literature review: Machine learning techniques applied to financial market prediction, Expert Syst. Appl. 124 (2019) 226–251.
[10] M. Ballings, D. Van den Poel, N. Hespeels, R. Gryp, Evaluating multiple classifiers for stock price direction prediction, Expert Syst. Appl. 42 (2015) 7046–7056.
[11] Y. Kara, M. Acar Boyacioglu, Ö.K. Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul stock exchange, Expert Syst. Appl. 38 (2011) 5311–5319.
[12] F. Zhou, H.-m. Zhou, Z. Yang, L. Yang, EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction, Expert Syst. Appl. 115 (2019) 136–151.

[13] J. Patel, S. Shah, P. Thakkar, K. Kotecha, Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques, Expert Syst. Appl. 42 (2015) 259–268.

[14] Y.-P. Huang, M.-F. Yen, A new perspective of performance comparison among machine learning algorithms for financial distress prediction, Appl. Soft Comput. 83 (2019) 105663.

[15] D.C.A. Mallqui, R.A.S. Fernandes, Predicting the direction, maximum, minimum and closing prices of daily Bitcoin exchange rate using machine learning techniques, Appl. Soft Comput. 75 (2019) 596–606.

[16] R. Dash, P.K. Dash, A hybrid stock trading framework integrating technical analysis with machine learning techniques, J. Finance Data Sci. 2 (2016) 42–57.

[17] D. Mahajan Shubhrata, V. Deshmukh Kaveri, R. Thite Pranit, Y. Samel Bhavana, P.J. Chate, Stock market prediction and analysis using Naïve Bayes, Int. J. Recent Innov. Trends Comput. Commun. 4 (2016) 121–124.

[18] I. Ghosh, R.K. Jana, M.K. Sanyal, Analysis of temporal pattern, causal interaction and predictive modeling of financial markets using nonlinear dynamics, econometric models and machine learning algorithms, Appl. Soft Comput. 82 (2019) 105553.

[19] M.H.D.M. Ribeiro, L. dos Santos Coelho, Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series, Appl. Soft Comput. 86 (2020) 105837.

[20] M. Zięba, S.K. Tomczak, J.M. Tomczak, Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction, Expert Syst. Appl. 58 (2016) 93–101.

[21] P. Carmona, F. Climent, A. Momparler, Predicting failure in the U.S. banking sector: An extreme gradient boosting approach, Int. Rev. Econ. Finance 61 (2019) 304–323.

[22] T.K. Lee, J.H. Cho, D.S. Kwon, S.Y. Sohn, Global stock market investment strategies based on financial network indicators using machine learning techniques, Expert Syst. Appl. 117 (2019) 228–242.

[23] B. Weng, M.A. Ahmed, F.M. Megahed, Stock market one-day ahead movement prediction using disparate data sources, Expert Syst. Appl. 79 (2017) 153–163.

[24] R. Bisoi, P.K. Dash, A.K. Parida, Hybrid variational mode decomposition and evolutionary robust kernel extreme learning machine for stock price and movement prediction on daily basis, Appl. Soft Comput. 74 (2019) 652–678.

[25] A.H. Moghaddam, M.H. Moghaddam, M. Esfandyari, Stock market index prediction using artificial neural network, J. Econ. Finance Adm. Sci. 21 (2016) 89–93.

[26] L.S. Malagrino, N.T. Roman, A.M. Monteiro, Forecasting stock market index daily direction: A Bayesian network approach, Expert Syst. Appl. 105 (2018) 11–22.

[27] A.N. Kia, S. Haratizadeh, S.B. Shouraki, A hybrid supervised semi-supervised graph-based model to predict one-day ahead movement of global stock markets and commodity prices, Expert Syst. Appl. 105 (2018) 159–173.

[28] W.-C. Chiang, D. Enke, T. Wu, R. Wang, An adaptive stock index trading decision support system, Expert Syst. Appl. 59 (2016) 195–207.

[29] B. Weng, W. Martinez, Y.-T. Tsai, C. Li, L. Lu, J.R. Barth, F.M. Megahed, Macroeconomic indicators alone can predict the monthly closing price of major U.S. indices: Insights from artificial intelligence, time-series analysis and hybrid models, Appl. Soft Comput. 71 (2018) 685–697.

[30] S. Barak, A. Arjmand, S. Ortobelli, Fusion of multiple diverse predictors in stock market, Inf. Fusion 36 (2017) 90–102.

[31] Y. Pan, Z. Xiao, X. Wang, D. Yang, A multiple support vector machine approach to stock index forecasting with mixed frequency sampling, Knowl.-Based Syst. 122 (2017) 90–102.

[32] R. Ramezanian, A. Peymanfar, S.B. Ebrahimi, An integrated framework of genetic network programming and multi-layer perceptron neural network for prediction of daily stock return: An application in Tehran stock exchange market, Appl. Soft Comput. 82 (2019) 105551.

[33] N. Oliveira, P. Cortez, N. Areal, The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices, Expert Syst. Appl. 73 (2017) 125–144.

[34] T.H. Nguyen, K. Shirai, J. Velcin, Sentiment analysis on social media for stock movement prediction, Expert Syst. Appl. 42 (2015) 9603–9611.

[35] X. Zhang, Y. Zhang, S. Wang, Y. Yao, B. Fang, S.Y. Philip, Improving stock market prediction via heterogeneous information fusion, Knowl.-Based Syst. 143 (2018) 236–247.

[36] N. Otter, M.A. Porter, U. Tillmann, P. Grindrod, H.A. Harrington, A roadmap for the computation of persistent homology, EPJ Data Sci. 6 (2017) 17.

[37] M. Kramár, R. Levanger, J. Tithof, B. Suri, M. Xu, M. Paul, M.F. Schatz, K. Mischaikow, Analysis of Kolmogorov flow and Rayleigh–Bénard convection using persistent homology, Physica D 334 (2016) 82–98.

[38] P. Dłotko, T. Wanner, Topological microstructure analysis using persistence landscapes, Physica D 334 (2016) 60–81.

[39] M. Nicolau, A.J. Levine, G. Carlsson, Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival, Proc. Natl. Acad. Sci. 108 (2011) 7265–7270.

[40] S. Emrani, T. Gentimis, H. Krim, Persistent homology of delay embeddings and its application to wheeze detection, IEEE Signal Process. Lett. 21 (2014) 459–463.

[41] J.A. Perea, A. Deckard, S.B. Haase, J. Harer, SW1PerS: Sliding windows and 1-persistence scoring; discovering periodicity in gene expression time series data, BMC Bioinformatics 16 (2015) 257.

[42] J.A. Perea, J. Harer, Sliding windows and persistence: An application of topological methods to signal analysis, Found. Comput. Math. 15 (2015) 799–838.

[43] M. Gidea, Topological data analysis of critical transitions in financial networks, in: E. Shmueli, B. Barzel, R. Puzis (Eds.), 3rd International Winter School and Conference on Network Science, Springer International Publishing, Cham, 2017, pp. 47–59.

[44] M. Gidea, Y. Katz, Topological data analysis of financial time series: Landscapes of crashes, Physica A 491 (2018) 820–834.

[45] M. Gidea, D. Goldsmith, Y. Katz, P. Roldan, Y. Shmalo, Topological recognition of critical transitions in time series of cryptocurrencies, Physica A (2020) 123843.

[46] D. Pachauri, C. Hinrichs, M.K. Chung, S.C. Johnson, V. Singh, Topology-based kernels with application to inference problems in alzheimer's disease, IEEE Trans. Med. Imaging 30 (2011) 1760–1770.

[47] V. Kovacev-Nikolic, P. Bubenik, D. Nikolić, G. Heo, Using persistent homology and dynamical distances to analyze protein binding, Stat. Appl. Genet. Mol. Biol. 15 (2016) 19–38.

[48] G. Muszynski, K. Kashinath, V. Kurlin, M. Wehner, Prabhat, Topological data analysis and machine learning for recognizing atmospheric river patterns in large climate datasets, Geosci. Model Dev. 12 (2019) 613–628.

[49] I. Obayashi, Y. Hiraoka, M. Kimura, Persistence diagrams with linear machine learning models, J. Appl. Comput. Topol. 1 (2018) 421–449.

[50] Z. Zhang, Y. Song, H. Cui, J. Wu, F. Schwartz, H. Qi, Early mastitis diagnosis through topological analysis of biosignals from low-voltage alternate current electrokinetics, in: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC, 2015, pp. 542–545.

[51] J. Yao, C.L. Tan, H.-L. Poh, Neural networks for technical analysis: a study on KLCI, Int. J. Theor. Appl. Finance 2 (1999) 221–241.

[52] W. Huang, Y. Nakamori, S.-Y. Wang, Forecasting stock market movement direction with support vector machine, Comput. Oper. Res. 32 (2005) 2513–2522.

[53] J.C.P. M'ng, A.A. Aziz, Using neural networks to enhance technical trading rule returns: A case with KLCI, Athens J. Bus. Econ. 2 (2016) 63–70.

[54] S.J. Abdulkadir, S.-P. Yong, M. Marimuthu, F.-W. Lai, Hybridization of ensemble Kalman filter and non-linear auto-regressive neural network for financial forecasting, in: Mining Intelligence and Knowledge Exploration, Springer, 2014, pp. 72–81.

[55] P.N. Bahrun, M.N. Taib, Selected Malaysia stock predictions using artificial neural network, in: 5th International Colloquium on Signal Processing & Its Applications, IEEE, 2009, pp. 428–431.

[56] V.D. Silva, G. Carlsson, Selected Malaysia stock predictions using artificial neural network, in: 5th International Colloquium on Signal Processing & Its Applications, IEEE, 2009, pp. 428–431.

[57] A. Cerri, B.D. Fabio, M. Ferri, P. Frosini, C. Landi, Betti numbers in multidimensional persistent homology are stable functions, Math. Methods Appl. Sci. 36 (2013) 1543–1557.

[58] Edelsbrunner, Letscher, Zomorodian, Topological persistence and simplification, Discrete Comput. Geom. 28 (2002) 511–533.

[59] R. Ghrist, Barcodes: The persistent topology of data, Bull. Amer. Math. Soc. 45 (2008).

[60] G. Carlsson, A. Zomorodian, A. Collins, L. Guibas, Persistence barcodes for shapes, in: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, ACM, Nice, France, 2004, pp. 124–135.

[61] H. Edelsbrunner, J. Harer, Persistent homology—a survey, Contemp. Math. 453 (2008) 257–282.

[62] Y. Umeda, Time series classification via topological data analysis, Inf. Media Technol. 12 (2017) 228–239.

[63] A. Zomorodian, G. Carlsson, Computing persistent homology, Discrete Comput. Geom. 33 (2005) 249–274.

[64] A.J. Zomorodian, Topology for Computing, Cambridge University Press, 2005.

[65] G. Carlsson, Topology and data, Bull. Amer. Math. Soc. 46 (2009) 255–308.

[66] B.T. Fasy, J. Kim, F. Lecci, C. Maria, Introduction to the R package TDA, 2014, arXiv preprint arXiv:1411.1830.

[67] C.-F. Tsai, Y.-C. Lin, D.C. Yen, Y.-M. Chen, Predicting stock returns by classifier ensembles, Appl. Soft Comput. 11 (2011) 2452–2459.

[68] M.-Y. Chen, Predicting corporate financial distress based on integration of decision tree classification and logistic regression, Expert Syst. Appl. 38 (2011) 11261–11272.
[69] C.-J. Lu, Integrating independent component analysis-based denoising scheme with neural network for stock price prediction, Expert Syst. Appl. 37 (2010) 7056–7064.
[70] A. Booth, E. Gerding, F. McGroarty, Automated trading with performance weighted random forests and seasonality, Expert Syst. Appl. 41 (2014) 3651–3661.
[71] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.
[72] F. Chazal, B. Michel, An introduction to topological data analysis: fundamental and practical aspects for data scientists, 2017, arXiv preprint arXiv:1710.04019.