


Review

A Study Concerning Soft Computing Approaches for Stock Price Forecasting

Chao Shi and Xiaosheng Zhuang * 

Department of Mathematics, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong;
stephen.shi@my.cityu.edu.hk

* Correspondence: xzhuang7@cityu.edu.hk

Received: 26 August 2019; Accepted: 16 October 2019; Published: 18 October 2019



Abstract: Financial time-series are well known for their non-linearity and non-stationarity nature. The application of conventional econometric models in prediction can incur significant errors. The fast advancement of soft computing techniques provides an alternative approach for estimating and forecasting volatile stock prices. Soft computing approaches exploit tolerance for imprecision, uncertainty, and partial truth to progressively and adaptively solve practical problems. In this study, a comprehensive review of latest soft computing tools is given. Then, examples incorporating a series of machine learning models, including both single and hybrid models, to predict prices of two representative indexes and one stock in Hong Kong's market are undertaken. The prediction performances of different models are evaluated and compared. The effects of the training sample size and stock patterns (viz. momentum and mean reversion) on model prediction are also investigated. Results indicate that artificial neural network (ANN)-based models yield the highest prediction accuracy. It was also found that the determination of optimal training sample size should take the pattern and volatility of stocks into consideration. Large prediction errors could be incurred when stocks exhibit a transition between mean reversion and momentum trend.

Keywords: machine learning; stock price prediction; Hong Kong's market; SVR; HMM; ANN; DWT; EMD

1. Introduction

Stock market is one of the most attractive and profitable investment arenas. The ability to predict stock movement can bring in enormous arbitrage opportunities, which are the main motivation for researchers in both academia and the financial industry. The classical efficient market hypothesis (EMH) states that stock price has reflected all the available information in the market and no one can consistently earn over the benchmark return solely based on the analysis of past stock price [1–3]. In other words, the movement of stock price follows a random walk and is totally unpredictable. However, evidences from various studies have shown that the market is not completely efficient [1,4,5], especially with the advent of computational intelligence technologies [6–8]. Various models and algorithms have been developed to exploit the market's inefficiency [9]. Cochrane [10] has claimed the predictability of the market return as the “New Facts in Finance.”

It is conventional to adopt econometric models (e.g., an autoregressive integrated moving average (ARIMA)) to predict stock prices. The rapid development in the speed and capability of commercial computers has prompted an emerging collection of methodologies, such as fuzzy systems, neural networks, machine learning, and probabilistic reasoning. Those methods collectively, are called soft computing, which aims to exploit tolerance for imprecision, uncertainty, and partial truth to progressively and adaptively solve practical problems [11]. Soft computing approaches have been successfully applied in various fields such as freight volume prediction [12], process control [13], and

wind prediction [14]. Most of the current non-linear and emerging machine learning models, such as the hidden Markov model, support vector machine, and artificial neural network, can be classified as soft computing models.

Note that classical statistical and econometric models, such as ARIMA and sophisticated autoregressive conditional heteroskedasticity (ARCH) models [15], are specifically developed for linear and stationary processes. A detailed comparison of different statistical models for stock price prediction and associated model details (e.g., data preprocessing, model input variables, model parameters, and performance measurements) has been summarized and documented by Atsalakis and Valavanis [16]. In addition, the application of above models for prediction may incur significant errors, as financial time-series are intrinsically non-linear, non-stationary, and exhibit high volatility. Actually, the role of statistical models in data analysis has been questioned by many scholars. It is criticized that approaching problems by searching for the most appropriate data model may restrict our imagination and cause a loss of accuracy and interpretability of the problem [17,18]. Therefore, people turn to non-linear, soft computing models for solving the dilemma. As the main component of soft computing is machine learning (ML), both terms will be used interchangeably from this point forward.

The machine learning approach for financial modeling is an attempt to find financial models automatically through progressive data's adaptation without any prior statistical assumptions [9]. The learning process can be considered as the mechanical process of acquiring a habit. Most ML models were created in 1980s and evolved slowly until the advent of super-powerful computers [6]. Those models can learn from complex data and adaptively extract optimal relationships between the input and output variables. The early success in stock market movement prediction and asset value estimation has further promoted its popularity [19]. ML itself is undoubtedly a powerful tool for data mining. However, the misapplication can also lead to disappointment. Arnott et al. [20] have, therefore, proposed a back-testing protocol for manipulating ML in quantitative finance.

Apart from the non-stationarity of financial time-series, another major challenge is multi-scaling [21], which refers to the fact that statistical properties of time-series change with time-horizons. To deal with the problem, various signal decomposition tools (e.g., discrete wavelet transform (DWT) and empirical mode decomposition (EMD)) have been borrowed from electrical engineering. Different hybrid methods incorporating the "divide-and-conquer" principle or "decomposition-and-ensemble" have been developed to predict the volatile time-series and obtained excellent performance [22]. The decomposition process can reduce the complexity of the time series and retrieve components of different frequencies and scales, enabling an improved prediction at specific time-horizons.

Despite the prevalent advantages, ML models have their own shortcomings. Some people criticize the fact that ML requires a large set of parameters. Others claim those models tend to suffer from overfitting. There is no consensus among financial experts regarding the predictive capacity of different models. Therefore, it is the primary objective of this research to compare the forecasting power of different ML models by a case study of two representative indexes and one stock in Hong Kong's market. Note that this study does not aim to develop trading strategies to earn extraordinary return.

The remainder of this study is organized as follows. Section 2 comprehensively reviews the commonly used "intelligent" ML methods. In Section 3, we give details of adopted ML algorithms for this study. The numerical experiment and numerical results are given in Section 4. Finally, conclusions regarding the applicability and accuracy of different models are drawn.

The contribution of this study is a comprehensive review of various ML approaches and their application in financial time-series prediction. In addition, different hybrid models and single ML models were utilized to predict two indexes and one stock in Hong Kong's market. Moreover, the effects of training sample size and stock trends on the prediction accuracy were evaluated and interpreted.

2. Literature Review

Atsalakis and Valavanis [23] surveyed the most common machine learning algorithms, including support vector machines (SVMs) and neural networks for financial time-series prediction. The recent popularity of intelligent technology has further expanded the family of neural networks to include fuzzy neural networks and other hybrid intelligent systems [24]. The application of hybrid machine learning in stock prediction has greatly improved the predicative capacity. In this section, a comprehensive review of the most fundamental single (i.e., SVM and artificial neural network (ANN)) and hybrid machine learning approaches (i.e., EMD and DWT based models) for financial time-series forecasting is carried out.

2.1. Hidden Markov Model

A process is defined as a Markov chain if its distribution of the next states does not depend on the previous state when the current state is given. For conventional Markov chain, the observations are visible and connected by state transition probabilities. In comparison, the hidden Markov chain has visible observations connected by hidden states. The multiple hidden states are assumed to follow multivariate Gaussian distribution. The probability of a certain observation is the product of the transition probability and emission probabilities.

The stock market can be viewed in the similar manner. The hidden states, which drive the stock market's movement, are invisible to the investors. The transition between different states is determined by company policy and economic conditions. For brevity, the emission probability between hidden states and output stock price is assumed to follow multivariate Gaussian distribution. This simplified assumption can induce errors when the stock exhibits strong skewness, which has been explicitly overcome by models accounting for skewness [25]. All the underlying information are reflected in the visible stock prices. Therefore, the hidden Markov model (HMM) can make a reasonable model for financial data.

Hassan and Nath [26] applied the principle to predict stock prices of airline companies and found that comparable prediction results with neural networks could be obtained. The rationale behind the HMM model is to locate the stock patterns from past stock price behavior, then statistically compare the similarity between neighboring price elements and make a prediction of tomorrow's stock price. Hassan [27,28] further extended the HMM model by integrating with fusion model (viz. ANN and genetic algorithms (GA)) and a fuzzy model to optimize the selection of input and output sequences. Later, maximum a posterior HMM [29] and extended coupled HMM models [30] were developed to predict stock prices in the USA and Chinese stock markets. Table 1 summarizes the abovementioned references.

Table 1. Summary of the hidden Markov model (HMM) based prediction methods used by this study as references.

Article	Stock Market	Input Variables	Training Method	Comparative Studies	Best Model
[26]	-	Opening, high, low and closing price	HMM	ANN	Similar performance
[28]	USA	Opening, high, low and closing price	ANN-HMM-GA	ARIMA	Similar performance
[27]	-	Opening, high, low and closing price	HMM-Fuzzy	HMM, HMM-ANN-GA, ARIMA and ANN	HMM-Fuzzy
[29]	USA	Fractional change, fractional high and low	Maximum a Posterior HMM	HMM-Fuzzy, ARIMA and ANN	HMM
[30]	China	Open, high, low and closing price; news articles	Extended coupled HMM	SVM, TeSIA, CMT, ECHMM-NE, ECHMM-NC, ECHMM	ECHMM

2.2. Support Vector Machine

A support vector machine (SVM) was originally developed for classification problems [31] and later extended for regression analysis (support vector regression, SVR). It can be viewed as a linear type model as it maps the input parameter into high-dimensional space for linear manipulation. The key step in applying the SVM is to find the support vectors and then maximize the margins between the vectors and the optimal decision boundary. Mathematically, the optimization problem finally evolves to a constrained optimization problem, which can be solved by the method of Lagrange multipliers.

With the introduction of an ϵ -sensitive loss function and versatile kernel functions, the capacity of SVM has been greatly extended to solve non-linear regression problems. The major advantage of SVM is that SVM implements a structural risk minimization principle (equal to the sum of conventional empirical risk minimization and a regularization term) [32], which is unlikely to suffer an overfitting problem [33]. It is found to have excellent generalization performance on a wide range of problems.

Table 2 shows a brief comparison of stock prediction by SVR based models. The application of SVM in financial time-series prediction was initiated by Tay and Cao [34]. They adopted SVM to forecast the return of five future contracts from the Chicago Mercantile Market and concluded that SVM outperformed a backpropagation neural network (BPNN). Similarly, Kim [33] used SVM to predict the Korean stock market. Twelve technical indicators were used as input variables and the optimal controlling parameters (viz. upper bound C and kernel parameter δ^2) were determined via trial and error. The prediction results were much better than those of its counterparts BPNN and the case-based reasoning method (CBR). In addition to stock price prediction, Huang et al. [35] utilized SVM to forecast the weekly movement of the Nikkei 225 index by correlating with macroeconomic factors (viz. exchange rate and Standard and Poor’s 500 Index (S&P500)). The prediction of up-to-minute stock price was carried out by Henrique et al. [36]; they concluded that the incorporation of a moving window could improve the predictability of the SVR. Recently, the SVM algorithm has been combined with other feature selection techniques (e.g., filter method and wrapper method) to improve the accuracy of trend prediction [37,38].

Table 2. Summary of support vector machine (SVM) based prediction methods used by this study’s references.

Article	Stock Market	Input Variables	Training Method	Comparative Studies	Best Model
[34]	USA	Daily closing price of five future contracts	SVM	BPNN	SVM
[33]	Korea	12 technical indicators	SVM	ANN and CBR	SVM
[35]	Japan	Weekly price of S&P500 Index and USD/Yen exchange rate	SVM	LDA, QDA, EPNN	SVM
[38]	USA	29 technical indicators and lagged index price	SVM	BPNN	SVM
[37]	Indonesia	14 technical indicators	Particle swarm optimization-SVR	-	-
[36]	Brazil, USA and China	5 technical indicators for daily and 1-minute price	SVR	Random walk model	SVR

2.3. Artificial Neural Network

Inspired by the human biological neural system (brain), the artificial neural network (ANN) was developed to mimic the processing mechanism of human neurons. The major advantage of ANN is its ability to learn from past experience and generalize underlying rules from past observations without any prior assumptions regarding the data’s distribution. It is well suited for solving problems whose embedding rules are hard to specify by explicit formulations. Moreover, it provides a universal function approximation for complex, non-linear and non-stationary processes [39]. It is not susceptible to the problem of model misspecification, which is common to statistical parametric models.

ANN-based models have gained wide acceptance in multiple disciplines, particularly for complicated pattern recognition and nonlinear forecasting tasks. Its application in the financial market, e.g., credit evaluation, portfolio management, and financial prediction and planning, is

detailed and summarized by Bahrammirzaee [24]. Atsalakis and Valavanis [23] also reviewed over 100 scientific publications and summarized the available neural and neuro-fuzzy techniques to forecast stock markets. The application of neural network in stock market forecasting was initiated by White [40] in 1988 and the constructed neural network failed to beat the efficient market hypothesis. Later, Kimoto et al. [41] employed a modular neural network for Tokyo’s exchange price index and a superior performance over conventional multiple regression analysis was concluded. Qiu and Song [42] adopted BPNN algorithms to forecast the trend of Nikkei225 based on two types of technical indicators filtered from the genetic algorithm (GA). The prediction accuracy reached as high as 81.3%. The successes of ANN when evaluating stock movement in Istanbul’s stock exchange and the Chinese market were reported by Kara et al. [43] and Cao et al. [44], respectively. Song et al. [45] predicted a weekly stock price from Chinese A stock by BPNN and concluded the results from BPNN outperformed other four neural network based models (viz. radial basis function neural network (RBFNN), SVR, least square-SVR (LS-SVR), and the general regression neural network (GRNN)). A time lagging effect between predicted and actual stock price was also observed. Table 3 shows a brief comparison of stock prediction by ANN based models.

Despite its advantages when dealing with volatile process, one of the major criticisms of ANN-based models is the interpretability of the model and the uncertainty of the selection of the control parameters (e.g., input variables, number of hidden layers, and hidden nodes), which are fully dependent on researcher’s prior experience [46].

Table 3. Summary of artificial neural network (ANN)-based prediction methods used by this study’s references.

Article	Stock Market	Input Variables	Training Method	Comparative Studies	Best Model
[40]	USA	Daily stock return	NN	-	Over-optimistic
[41]	Japan	Technical indicators and economic indexes	NN	Multiple regression analysis	NN
[44]	China	Daily closing price; quarterly book value; common share outstanding	ANN	CAPM, Fama and Frech’s 3 factor model	ANN
[43]	Turkey	10 technical indicators	ANN	SVM	ANN
[47]	USA	Daily opening, high, low and closing price; trading volume	ANN	ARIMA	ANN
[42]	Japan	Technical indicators	GA-ANN	-	-
[45]	China	Weekly close price	BPNN	RBF, GRNN, SVR, LS-SVR	BPNN

2.4. Discrete Wavelet Transform-Based Models

Discrete wavelet transform (DWT) as well as its generalization [48–51], are commonly used for image procession [52,53], earthquake prediction, and so forth [51,54]. It gained popularity in the economic and finance industries due to its capability of revealing localized information for data in the time domain. The method provides an alternative perspective to inspect the relationship in economics and finance [4,55]. More specifically, the method can decompose a discrete time sequence into binary high and low-frequency parts. The high-frequency component can capture the local discontinuities, singularities, and ruptures [56]. On the other hand, the low frequency delineates the coarse structure for the long-term trend of the original data. The combination of high and low frequency enables the extraction of both hidden and temporal patterns of the original time-series. By working with the decomposed parts, it is conceivable that the future stock of each part can be better predicted due to the much less inherent variation of each component.

For most studies, DWT has been treated as a denoising tool to eliminate redundant variables and integrate with other modelling tools for prediction [57–59]. Li et al. [57] derived nine alternative technical indicators based on coarse coefficients to predict the direction of the Down Jones Industrial Average (DJIA). The results revealed that wavelet analysis could help extract meaningful input variables for improving the forecasting performance of a genetic programming algorithm. Apart from feature selection, the direct application in stock price prediction has also been performed by various scholars. Jothimani et al. [60] applied a hybrid method to predict the weekly price movement of the National

Stock Exchange Fifty index. The stock price was first decomposed by enhanced maximum overlap discrete wavelet transform (MODWT) and each component was predicted by SVR and ANN methods. The simulation results indicated the hybrid method could improve the directional accuracy from merely over 50% to about 57%. However, it should be noted that the stock price studied exhibits a momentum pattern. The performance for stocks exhibiting mean reversion pattern are not known. Chandar et al. [61] adopted DWT approach to decompose stock prices of five different companies into approximation and detail components. The decomposed components were combined with other variables (e.g., volume) to serve as input parameters for BPNN. It was found that the combined models yielded much smaller errors compared with the conventional BPNN model. Table 4 shows a brief summary of stock prediction by DWT.

Table 4. Summary of discrete wavelet transform (DWT)-based prediction methods used by this study’s references.

Article	Stock Market	Input Variables	Training Method	Comparative Studies	Best Model
[62]	-	Weekly exchange rate GBP/USD	DWT-ANN/ARIMA	ARIMA, ANN and Zhang’s hybrid model	DWT-ANN/ARIMA
[60]	India	Weekly closing price	MODWT-ANN/SVR	ANN, SVR	MODWT-ANN/SVR
[56]	USA	Minute-in-day closing price	DWT-BPNN	ARMA, Random walk model	DWT-BPNN
[59]	China	Monthly closing price	DWT-BPNN	BPNN	DWT-BPNN
[58]	Taiwan, USA, UK, Japan	10 technical indicators	DWT-ABC-RNN	DWT-BP-ANN, BNN	DWT-ABC-RNN
[57]	USA	9 technical indicators	DWT-FGP	-	-

2.5. Empirical Mode Decomposition-Based Models

Empirical mode decomposition (EMD) was invented by Huang et al. [63] and it has been widely used in many fields, such as earthquake analysis and structure analysis [64]. The major advantage of EMD is the ability to adaptively represent a nonstationary signal as the summation of a series of zero-mean-amplitude modulation components. It is possible to reveal hidden patterns and identify potential trends for forecasting [65].

The rationale of EMD is to decompose a complicated time-series process into multiple simpler orthogonal low-frequency components (viz. intrinsic mode function (IMF)) with strong correlations, rendering a more accurate prediction. The reconstruction of the original process is enabled by the addition of all the IMFs and the residual component. Each IMF has retained local characteristics of the original signal and could be better extracted and analyzed.

Drakakis [66] was initiated to explore the feasibility of using EMD to study the stationarity of Dow-Jones’ volume. He found that the decomposed IMFs centered around zero and were some kind of posterior orthonormal wavelet basis. At the same time, Nava et al. [67] applied a combined EMD and SVR algorithm to predict the multistep ahead price of S&P 500. The combined model was found to outperform benchmark models (i.e., SVR). Particularly, the high-frequent IMFs were better to forecast short-term performance. In contrast, the long-term behavior could be well captured by the low-frequency IMFs. Similarly, a combined EMD and SVR method to forecast Taiwan stock exchange capitalization weighted index (TAIEX) was undertaken by Cheng and Wei [68]. Superior prediction results were obtained over classic SVR and autoregressive (AR) models. Lin et al. [69] implemented EMD and the least squares-SVR method to estimate three foreign exchange rates and better performance was obtained. EMD was also combined with the ANN model to predict the volatility of the Baltic dry index (BDI) [70]; the forecasting results were found to be reasonably better than pure SVR and ANN models. Table 5 shows a brief comparison of stock prediction by EMD.

Table 5. Summary of empirical mode decomposition (EMD)-based prediction methods used by this study’s references.

Article	Stock Market	Input Variables	Training Method	Comparative Studies	Best Model
[22]	USA, UK	Daily crude oil spot price	EMD + ARIMA + ALNN; EMD-FNN-ALNN; EMD-ARIMA-Average	ARIMA, FNN	EMD - FNN - ALNN
[71]	China	Closing price	EMD-SVM	SVM	EMD-SVM
[69]	-	Daily exchange rate USD/NTD, JPY/NTD and RMB/NTD	EMD-LSSVR	EMD-ARIMA, LSSVR, ARIMA	EMD-SVR
[68]	Taiwan	Closing price	EMD-SVR	AR, SVR	EMD-SVR
[65]	Taiwan, HK	Closing price	EMD-ANFIS	SVR, AR, ANFIS	EMD-ANFIS
[67]	USA	Intraday price	EMD-SVR	ARIMA, Directed SVR, Recursive SVR	EMD-SVR

3. Data and Methods

3.1. Data Sets

The application of various soft computing techniques for analyzing USA’s markets and other liquid markets have been studied extensively. For those mature markets, institutional investors dominate the market and determine the overall trend. There is limited research specifically for Hong Kong’s market. The peak trading value of Hong Kong’s market is about one hundred billion Hong Kong Dollar (HKD) per day, half of which is contributed by personal investors. Therefore, it is worthwhile to have a detailed study of the applicability of the abovementioned machine learning models specifically in Hong Kong’s market.

In this study, two indexes and one stock, namely the Hang Seng Index (HSI), Hang Seng China Enterprises Index (HSCEI) and Tencent, were taken for the numerical experiments. The choice of the three time series was based on the consideration of liquidity and representativeness of Hong Kong’s market. For all the three stocks, the past ten years’ daily closing prices between March 9, 2009 and March 8, 2019 downloaded from Yahoo finance served as the input variables. There are 2474 daily closing prices in total. For the dates with no stock price data, interpolation was used to deduce the appropriate values. It should be noted that interpolated values only account for 0.4% of all the prices and should not affect the results and the conclusions of this study. Figure 1 shows the time plots of the three stocks. It is evident that Tencent shows a strong, monotonic, increasing trend before 2018. In contrast, the pattern for both HSI and HSCEI tend to be recurrent throughout the whole period. Table 6 shows the calculated basic statistics (i.e., mean, standard deviation, and skewness) of the daily returns of the three chosen stocks. Specifically, the daily return is defined as the percentage of a stock’s price change over the previous day’s closing price. It should be noted that skewness can have a major influence on portfolio construction [72] and model selection [25], which has been the interest of many researchers.

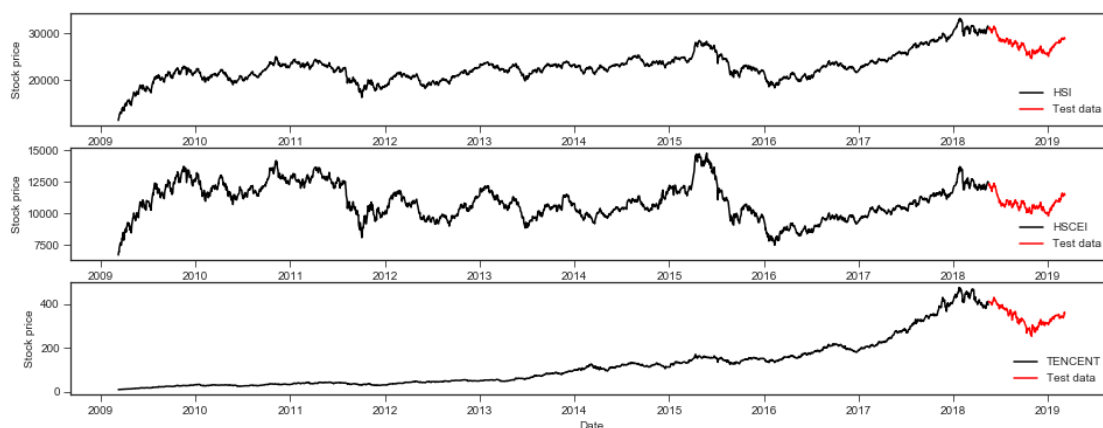


Figure 1. Plots showing the daily close prices between March 9, 2009 and March 8, 2019.

Table 6. Basic statistics of daily return of the three stocks.

Statistics ¹	Hang Seng Index	Hang Seng China Enterprises Index	Tencent
Mean (μ)	0.05%	0.04%	0.19%
Standard Deviation (σ)	1.24%	1.54%	2.07%
Skewness (S_k)	0.01	0.10	0.14
Kurtosis (β)	2.51	1.98	2.04

Note: ¹ $\mu = \frac{\sum_{i=1}^N r_i}{N}$, $\sigma = \sqrt{\frac{\sum_{i=1}^N (r_i - \mu)^2}{N}}$, $S_k = \frac{1}{N} \frac{\sum_{i=1}^N (r_i - \mu)^3}{\sigma^3}$, and $\beta = \frac{1}{N} \frac{\sum_{i=1}^N (r_i - \mu)^4}{\sigma^4}$ where r and N stand for daily return and population number, respectively.

It should be noted for all the numerical simulations, the test dataset was fixed at the latest 200 days' closing prices. All the other data were taken for training to determine the optimal weights for different ML models. The dynamic training window size was fixed at five, which means the next-step stock price was predicted by the pre-determined weighted sum of past 5 days' stock price. The determination of 5 day was based on a partial autocorrelation plot (PACP) and published literature [60]. It should be noted that only one-step ahead prediction was carried out in this study.

3.2. Comparison Criteria

Forecasting error measures can be based on absolute, percentage, symmetric, relative, scaled, and other errors [73]. Each performance measure evaluates forecasting error from a different aspect. As suggested by Shcherbakov et al. [73], absolute error (e.g., mean absolute error) is recommended for comparing time-series predictions. To remove scale dependency, the absolute error can be transformed into absolute percentage error. In this study, the predictive power of the forecasting models was evaluated by absolute percentage error (APE) and mean absolute percentage error (MAPE). APE is defined as the absolute percentage difference between actual stock price and predicted value at a given day i . MAPE is the average of all the calculated APEs. The detailed formulation is defined as follows [74]:

$$APE = \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

where n is the total number of the test data points; y_i and \hat{y}_i are the actual price and predicted stock price on day i .

3.3. HMM

The rationale behind the HMM model is to locate the stock patterns from past stock price behavior, then statistically compare the similarities between neighboring price elements and make predictions of tomorrow's stock price [26]. The price differences between the matched segments and the next day are deduced.

For stock price prediction, the priority is to calculate the likelihood of current observation sequence and try to identify the most similar sequence in the history. Then, the future stock price can be predicted based on the next step price from history sequence. While in reality, the hidden states are invisible and the transmission probabilities are not readily available.

The main purpose of applying HMM is to estimate the optimal parameters (i.e., mean and variance for GMM, transmission, and emission probabilities) and identify the similar sequence in history. The number of hidden states was allowed to vary between two and five, with the optimal value determined by Akaike information criterion (AIC) [75]. The goal can be achieved by the combination of three algorithms (viz. forward algorithm, Viterbi algorithm and EM algorithm) [76].

3.4. SVM

SVM is developed specifically for dealing with linearly non-separable problems by classical regression methods. The ideal is to transform the input from original space into a high-dimensional feature space via a mapping function. The problem then becomes linear regression or classification over new feature space. The optimal regression function is formulated as follows:

$$y = \omega^T \cdot \varphi(x) + b,$$

where φ is the mapping function to transform the input variable; ω and b are the coefficients estimated by the following constrained quadratic minimization problem [77]:

Minimize

$$\frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*),$$

subject to

$$y_i - \omega \cdot x_i - b \leq \varepsilon + \xi_i,$$

$$\omega \cdot x_i + b - y_i \leq \varepsilon + \xi_i^*,$$

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

where y_i is the *i*stock price; C denotes penalty parameter; ξ_i and ξ_i^* are slack variables to cater for data lying outside bandwidth ε . The minimization problem can be converted to dual representation after applying Lagrangian and optimality conditions:

$$y = \sum_{i=1}^n (\alpha_i - \alpha_i^*) k(x_i, x) + b,$$

where α_i and α_i^* are Lagrangian multipliers and the kernel function $k(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$ for the radial basis function (RBF) used in this study. The advantage of RBF is its flexibility in handling diverse input sets and weighing closer training points more heavily [78]. This is most appropriate for predicting stock prices with trends. The performance of SVR depends on the selection of the model parameters C and σ , both of which are determined via 5-fold cross validation by grid searching among {1, 10, 100, 1000} and {1e-2, 1e-1, 1, 1e1, 1e2} for C and σ , respectively.

3.5. ANN

Among all the neural networks, the BP algorithm is the most widely adopted classical learning method. A typical three-layer BP neural network consisting of an input layer, hidden layer, and output layer is utilized in this study. The neurons between consecutive layers are connected via weight functions. The information from the input layer transfers to the output by weighted sum of the input variables. The estimated output is then compared with the actual value and errors are backpropagated via the network for updating weights and biases repeatedly. Figure 2 shows the graphical representation of the BPNN. For the general point of view, the neural network can reveal any hidden patterns, given sufficient training data.

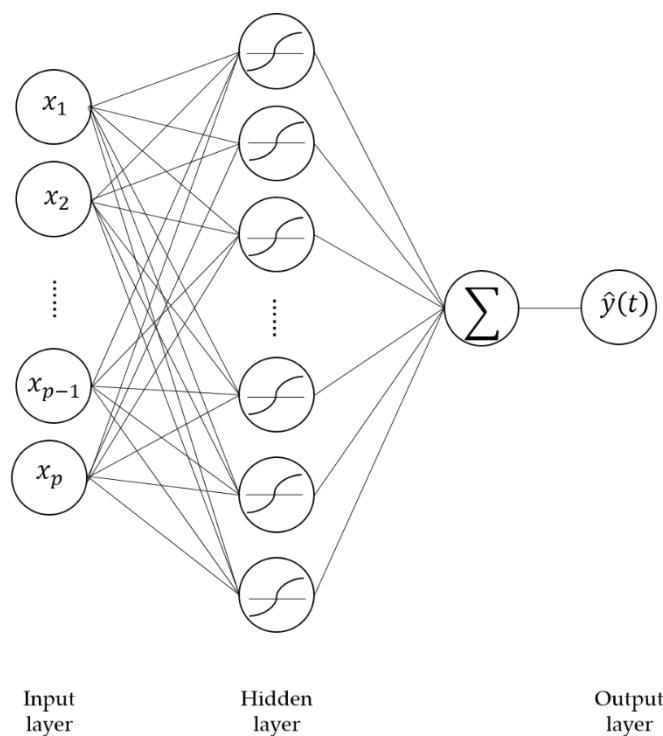


Figure 2. Graphical representation of the typical three-layer backpropagation neural network.

Each hidden neuron performs an operation of weighted summation. Mathematically, the output at the *i*th neuron of the hidden layer can be formulated as follows:

$$h_i = \sigma\left(\sum_{j=1}^p \omega_{i,j}x_j + b_i\right),$$

where σ is the activation function and is taken to be the sigmoid function; $\omega_{i,j}$ is weight coefficient associated with input x_j from previous layer; b_i denotes the bias; and p represents the number of input variables. For a single output, as shown in Figure 2, its final output is calculated as follows:

$$\hat{y}(t) = b_o + \sum_{i=1}^q \omega_i \cdot h_i,$$

where $\hat{y}(t)$ is the predicted stock price, q is the number of hidden neurons, b_o and ω_i are the bias and weight, respectively. The error term for the backpropagation is estimated via mean sum error (MSE):

$$MSE = \frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2,$$

where y_t is the actual stock price and n denotes the number of test samples.

In this study, the Keras sequential model [6] was adopted as the modelling tool. All the input stock prices were normalized by the maximal stock price before feeding into the network. The determination of the optimal number of hidden neurons was based on past experience via a trial-and-error manner. There are some guidelines [79] in the literature regarding the selection of the optimal number of hidden layers. In this study, a pruning approach method was adopted. The optimal number was assumed to be 256 and was successively reduced by half until a satisfactory prediction was obtained. In this study, the one-hidden-layer network with 32 nodes was selected. The batch size and epoch were chosen to be 50 and 128, respectively. The determination of epoch and size of layer is achieved via check of

validation loss. It is noted that by the use of abovementioned parameters, there was no overfitting in the validation process.

3.6. DWT

Wavelet analysis overcomes the drawback of conventional Fourier transformation by introducing more flexible basis functions (e.g., Haar basis) and scaling functions, enabling the multiresolution representation of a time-series. DWT is specifically for dealing with discrete processes. DWT is capable of identifying and locating local variations within time domain. A typical dyadic DWT successively decomposes the original signal into high and low frequency components with the sequential ratio reduced by half. Detailed wavelet analysis can refer to [51]

$$A_{s,k} = \int y(t) \cdot \Phi_{s,k}(t) dt, D_{s,k} = \int y(t) \cdot \Psi_{s,k}(t) dt,$$

where $A_{s,k}$ and $D_{s,k}$ are approximation and detail coefficients; s and k denote scaling and translation parameters, respectively; Φ and Ψ represent father and mother wavelets, approximating the smooth (low frequency) and the detailed (high-frequency) components. Father and mother wavelets are defined as follows:

$$\Phi_{s,k}(t) = 2^{-s/2} \cdot \Phi(2^{-s}t - k), \Psi_{s,k}(t) = 2^{-s/2} \cdot \Psi(2^{-s}t - k),$$

After deriving all the coefficients, the original signal can be reconstructed as the sum of a series of orthogonal wavelets:

$$y(t) = \sum_k A_{s,k} \cdot \Phi_{s,k}(t) + \sum_k D_{s,k} \cdot \Psi_{s,k}(t) + \sum_k D_{s-1,k} \cdot \Psi_{s,k}(t) + \dots + \sum_k D_{1,k} \cdot \Psi_{1,k}(t).$$

The level of decomposition was set to two in this study. An increase in level may possibly lead to the loss of information. The popular Daubechies [80] wavelet was selected as the desired decomposition wavelet.

3.7. EMD

EMD method can decompose a noisy signal into multiple IMFs and the residual through sifting process. EMD is regarded as a data-adaptive processing technique, which has never been fully appreciated to its full potential of revealing the truth and extracting the coherence hidden in scattered numbers [64]. The generated IMFs represent fast to low frequency oscillations and the residual component denotes the mean trend of the time series data. The integration of all the oscillation components synthesizes the original signal.

Given a signal $y(t)$, the sifting process for finding the IMF sifting procedures for the decomposition is summarized as follows

- (1) Identify all the local maxima and minima of $y(t)$.
- (2) Connect all the local maxima and local minima separately by the cubic spline to form the upper envelope line $h(t)$ and lower envelope line $l(t)$.
- (3) Calculate the mean value of the upper and lower envelope $m(t) = (h(t) + l(t))/2$.
- (4) Derive a new time-series $c(t)$ by subtracting the mean envelope, $c(t) = y(t) - m(t)$.
- (5) If $c(t)$ satisfies the properties of IMF [64]; then, $c(t)$ is regarded as an IMF, and $y(t)$ in step 1 is replaced with the new process $r(t) = y(t) - c(t)$. Otherwise, substitute $y(t)$ in step 1 by $c(t)$ and repeat all of the above process.

When the specified number of IMFs is obtained, the sifting process comes to an end. Similarly, the original signal can be reconstructed by the aggregation of all the IMFs and the residual as follows.

$$y(t) = \sum_{i=1}^N c_i(t) + r_N(t).$$

In this study, all the signal decomposition processes were performed with the PyEMD package in Python 3.6.

3.8. Overall Process of Hybrid Machine Learning Models

For each given stock time-series, it was first preprocessed by DWT and EMD to extract multiple components of high and low frequency. Each decomposed part was then independently predicted by ANN and SVR and aggregated to obtain the final forecasts. The below flowcharts, namely Figures 3 and 4, detail the modelling procedures for DWT and EMD based hybrid models for stock price prediction, respectively.

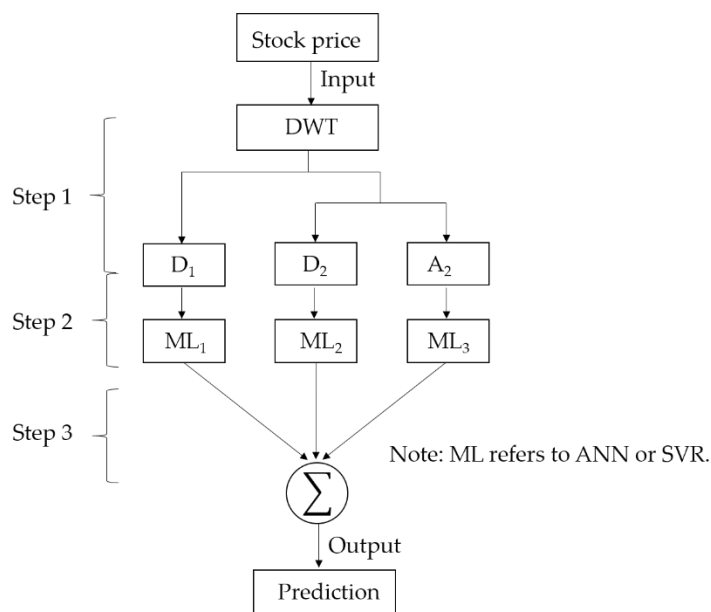


Figure 3. Overall process of the DWT-based model.

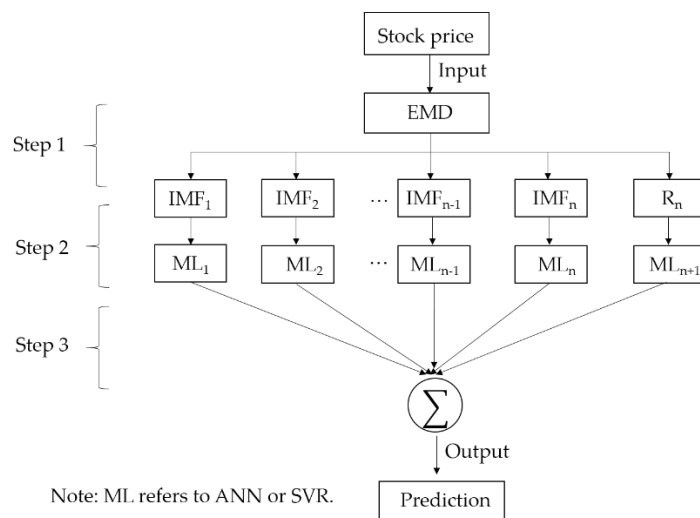


Figure 4. Overall process of the EMD-based model.

4. Experimental Results

In this study, all the numerical simulations were performed by various packages (i.e., hmmlearn, sklearn, keras, PyEMD, and PyWavelets) in Python 3.6. Firstly, experiments were separately conducted by HMM, SVR and ANN. For the hybrid method, the “divide-and-conquer” principle was implemented [22]. Each time-series was decomposed by DWT and EMD before integrating with SVR and ANN for further forecasting. The final prediction value was obtained by aggregating all the predictions.

4.1. MAPE Comparison

Table 7 summarizes the MAPE for one-day ahead prediction by different models. For reference, the standard deviation of daily return of the three stocks is shown in Table 6. It is noted that Tencent has the largest standard deviation (S.D.) for a daily return of 2.07%, compared with 1.24% and 1.54% for HSI and HSCEI, respectively. This can be explained by the fact that HSI and HSCEI are market indices which made up of many low volatile companies from different sectors.

Table 7. Comparison of mean absolute percentage errors (MAPEs) for different models.

Methodology	Hang Seng Index		Hang Seng China Enterprises Index		Tencent	
	MAPE (%)	Rank	MAPE (%)	Rank	MAPE (%)	Rank
Benchmark S. D.	1.24	-	1.54	-	2.07	-
Random walk	0.97	-	1.13	-	1.68	-
HMM	1.42	4	1.27	3	2.41	4
SVR	1.69	6	1.86	6	2.70	5
DWT - SVR	1.85	7	2.04	7	3.10	7
EMD - SVR	1.45	5	1.57	5	2.60	6
ANN	1.12	2	1.04	1	1.81	1
DWT - ANN	1.11	1	1.20	2	1.83	2
EMD - ANN	1.21	3	1.35	4	2.25 ¹	3

Note: ¹ MAPE for Tencent was obtained using a fixed training sample size of 200.

As for the forecasting of HSI, ANN based models yielded the smallest MAPEs. More specifically, hybrid DWT-ANN and single ANN algorithms outperformed that of EMD-ANN. That may imply that DWT can help decompose the original time-series into meaningful subcomponents, facilitating the prediction of ANN. Further improvement could be obtained by DWT-ANN if different combinations of wavelets and decomposition levels were utilized. In comparison, the results are not so encouraging for SVR based models. Although the optimal hyperplane parameters for SVR were determined by cross-validation, the MAPE for those models was still larger than 1.45%. This result is different from previous studies [33,38], which have reported that the SVR model outperforms ANN models. Note that the input variables of those models involve technical indicators, which are derived from both trading price and volume. Intuitively, the prediction performance can be improved if more market information is incorporated. Another point to note is that SVR performance can be improved if a self-adaptive EMD method is used to decompose the stock price first. This may imply that the underlying information embedded within the stock price has not been fully exploited by SVR. Moreover, the prediction MAPE for HMM model lies in between ANN and SVR based models. Of all the above mentioned results, only ANN based models have smaller prediction errors than the benchmark daily standard deviation (S.D.) of HSI itself (see Table 7). Overall, it implies that the stock market is indeed predictable.

Similar results were also obtained for HSCEI and Tencent. The prediction results from ANN based models surpassed both HMM and SVR based models. More importantly, DWT based ANN models consistently outperformed that of EMD based ANN models, particularly for stocks exhibiting higher fluctuations. In addition, it is observed that time-series preprocessed by EMD can be better predicted by SVR for all the three stocks than DWT based models. This can be explained by the fact

that sub-series of DWT are derived by convolution with a pre-determined basis function (viz. wavelet). The performance of DWT-SVR may possibly depend on the choice of wavelet. It is also worthwhile to note that HMM model has a smaller MAPE (1.27%) to the S.D. of HSCEI. As the rationale of the HMM algorithm is to locate the similar observation sequence from historical data for one-step ahead prediction, HMM works best for stocks with recurrent patterns. This is reflected in Figure 1, where HSCEI mainly fluctuates between 10,000 and 12,500. It is noted that the MAPE for random walk, which assumes the best prediction for the tomorrow's stock price equals yesterday's price, is also included for comparison. It can be seen only the ANN prediction for HSCEI beats the result of random walk. It may imply that prediction purely based on stock price can hardly beat the efficient market hypothesis. More stock and market information (e.g., trading volume and market sentiment) should be incorporated to strengthen the prediction capability of soft computing techniques.

4.2. Sample Size Effect

One of the major criticism of ANN is that the calculation tends to be trapped by local extrema during training process and requires lots of training parameters [46]. The significance of sample size determination has often been neglected. In this study, we took all the historical data beyond the latest 200-day closing price as the training sample. The total size of training sample was over 2000, which was used to optimize the weights of the 32 hidden neurons. A hidden layer with 32 neurons is considered enough to approximate any forms of functions. When applying the hybrid method, namely EMD-ANN, to predict the stock price of Tencent, we observed the performance was quite poor, even though a series of neural network parameters (i.e., number of hidden layers, number of neural nodes, activation function, and epoch number) were tuned. Finally, we reduced the sample size from over 2000 to 200. The performance was greatly improved, with the MAPE reducing from 4.10% to 2.25%. Figure 5 shows the comparison of EMD+ANN models utilizing different sample sizes.

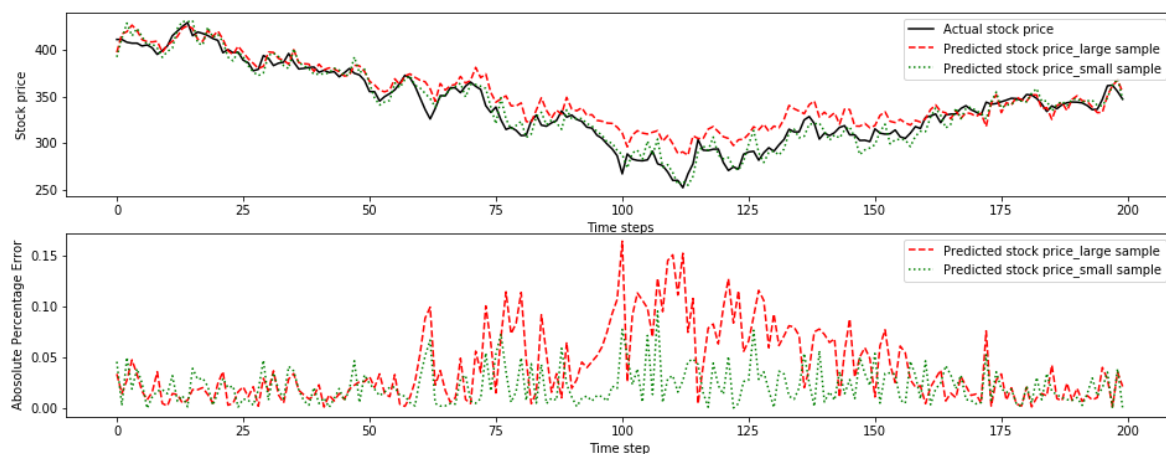


Figure 5. Performance of Tencent by EMD + ANN over different sample sizes.

The large deviation of predicted and actual stock price for a larger sample size can be explained by the fact that Tencent exhibits a monotonic increasing trend before 2018 (see Figure 1). The training over those samples adds weight to the belief that the stock will continue to monotonically increase, leading to worse forecasting for the recent large drawdown. In essence, it is overfitting to the past stock price [46]. It is noted that the same phenomenon does not apply to both HSI and HSCEI, possibly due to the more recurrent stock pattern and smaller volatility. For this particular case, the importance of sample size seems to overweight the combination of different neural network parameters. A rationale method to determine the optimal sample size taking stock patterns and stock volatility into consideration should be developed.

4.3. Momentum and Mean Reversion Stock Pattern Effect

Previous studies have mainly focused on performance comparisons of different algorithms but cast little light on their applicability and interpretation. In this section, the prediction performance over different time horizons and stock patterns are compared in detail. Tencent was selected for the comparison, as it exhibited the highest volatility of all the three stocks (see Table 6).

Mean reversion is defined as stock return exhibiting negative auto-correlation at certain horizons [81]. In contrast, momentum describes the continuation of stock return trend [82]. In this study, three trading strategies, namely positive momentum, negative momentum, and mean reversion, were developed to detect the stock trend. More specifically, positive momentum adopts the strategy that if the previous day’s return was positive then stock is to be bought and held on the next day. On the contrary, negative momentum denotes shorting the stock if a negative return occurred the previous day. While mean reversion represents buying and holding the stock if the previous day’s return was negative.

Figure 6 shows the cumulative return of Tencent utilizing the three trading strategies. It is clear that negative momentum dominates the first 100 time steps and an increasing trend of mean reversion strategy is prevalent after 125 time steps. The time plots of APE from all the algorithms are shown in Figure 7. Clearly, no differences in terms of APE can be observed irrespectively of whether momentum or mean reversion dominates. Another salient point needing to be noted is that all the algorithms show a bad performance during the transition period between momentum and mean reversion. That implies that algorithms capable of detecting market turning points should be integrated when developing prediction algorithms [83].

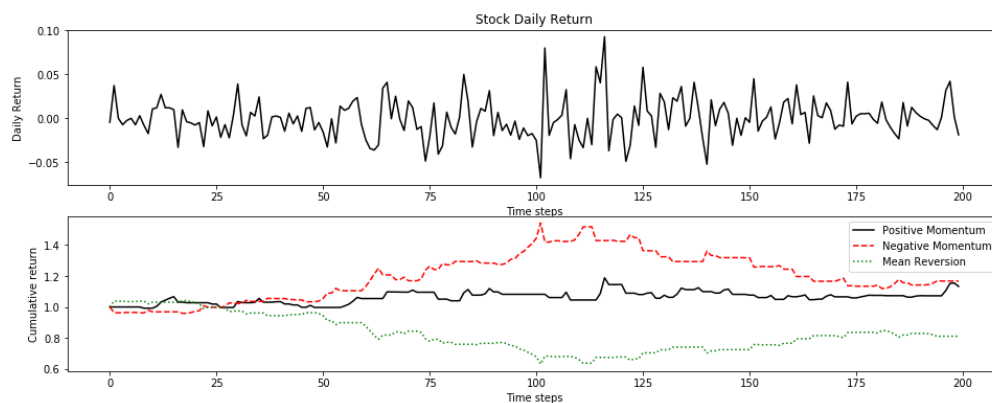


Figure 6. Daily stock return and stock pattern of Tencent.

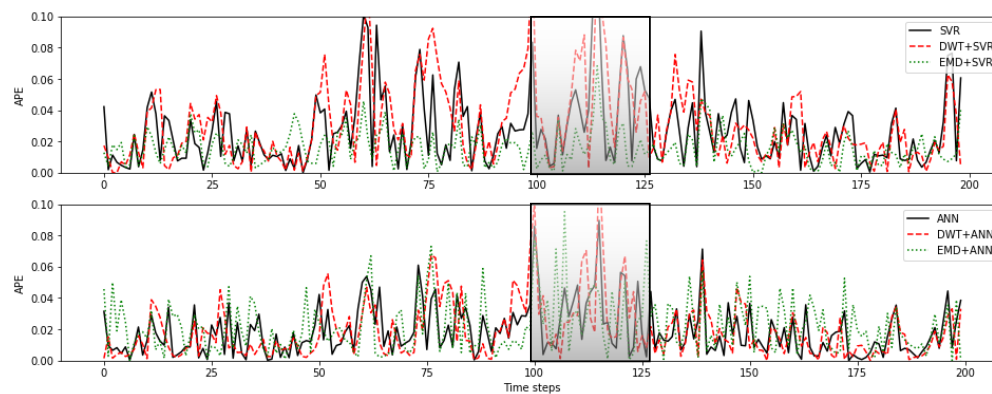


Figure 7. MAPE of different models for Tencent.

5. Conclusions

This study investigated and compared the performance of different machine learning algorithms; namely, the hidden Markov model (HMM), support vector machine (SVM), and artificial neural network (ANN). In addition, discrete wavelet transform (DWT) and empirical mode decomposition (EMD) methods were also used to preprocess time-series data and each subsequence was then integrated with SVR and ANN to predict the performance of three stocks; viz., the Hang Seng Index (HSI), Hang Seng China Enterprises Index (HSCEI), and Tencent.

The results indicate that ANN based models yield smaller MAPEs to those from HMM and SVR based models. More importantly, comparable prediction performances could be obtained by DWT-ANN and ANN algorithms for stocks with low and high variations. This may imply that DWT could decompose original time-series into meaningful subcomponents to facilitate the prediction of ANN. Further improvements could possibly be achieved if different wavelets and decomposition levels are employed. In contrast, the MAPE predicted by the SVR model can be further reduced when combined with EMD methods, as EDM can adaptively decompose stock price into multiple 'pure' parts without any prior basis. In addition, HMM calculates the MAPE in between ANN and SVR based models. Of all the predictions, only ANN based models can have smaller MAPEs than the intrinsic S.D. of an individual stock price.

Factors affecting the forecasting performance of hybrid ANN models are also investigated. It is noted that the prediction into the future depends on the choice of training sample size. For Tencent, it exhibited momentum trends over the past 10 years; the recent mean revert patterns could not be well captured if the model trained over the whole 10-year stock price. Thus, it is critical to select a training dataset when there is a change in the stock pattern. In addition, the performance of different machine learning models for Tencent over momentum and mean reverting patterns were also compared. Results indicate that the adopted models work well during both momentum and mean reverting stock patterns. However, during the transition period between different stock patterns, the performances of all the hybrid models were quite poor.

It should be noted that the purpose of this study is to demonstrate the forecasting capacity of common machine learning models. The decomposition methods, namely DWT and EMD, still have great room for further improvement, such as the optimal feature selection and special handling of large local variations. Tailor-made, specific forecasting techniques could be employed for forecasting different time-scale subseries, and thereby gain an improvement in prediction.

Author Contributions: Both authors participated equally in this work.

Acknowledgments: The research of C.S. and X.Z. was supported in part by the Research Grants Council of Hong Kong (project number CityU 11300717).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

AIC	Akaike Information Criterion
ANFIS	Adaptive Neuro Fuzzy Inference System
ANN	Artificial Neural Network
APE	Absolute Percentage Error
AR	Autoregressive Model
ARCH	Autoregressive Conditional Heteroskedasticity Model
ARIMA	Autoregressive Integrated Moving Average

BDI	Baltic Dry Index
BP	Backpropagation
BPNN	Backpropagation Neural Network
CAPM	Capital Asset Pricing Model
DJIA	Down Jones Industrial Average
DWT	Discrete Wavelet Transform
EM	Expectation-Maximization Algorithm
EMD	Empirical Mode Decomposition
EMH	Efficient Market Hypothesis
GA	Genetic Algorithm
GMM	Gaussian Mixture Model
GRNN	General Regression Neural Network
HMM	Hidden Markov Model
HSCEI	Hong Seng China Enterprises Index
HSI	Hang Seng Index
IMF	Intrinsic Mode Function
LDA	Linear Discriminant Analysis
LS-SVR	Least Square Support Vector Regression
MAPE	Mean Absolute Percentage Error
MODWT	Maximum Overlap Discrete Wavelet Transform
ML	Machine Learning
NN	Neural Network
RBF	Radial Basis Function
RBFNN	Radial Basis Function Neural Network
S.D.	Standard Deviation
SVM	Support Vector Machine
SVR	Support Vector Regression

References

1. Malkiel, B.G. *A Random Walk down Wall Street: Including a Life-Cycle Guide to Personal Investing*, 7th ed.; WW Norton & Company: New York, NY, USA, 1999.
2. Malkiel, B.G.; Fama, E.F. Efficient capital markets: A review of theory and empirical work. *J. Financ.* **1970**, *25*, 383–417. [[CrossRef](#)]
3. Fama, E.F. Efficient capital markets: II. *J. Financ.* **1991**, *46*, 1575–1617. [[CrossRef](#)]
4. Ramsey, J.B. Wavelets in economics and finance: Past and future. *Stud. Nonlinear Dyn. Econom.* **2002**, *6*, 1–27. [[CrossRef](#)]
5. Atsalakis, G.S.; Valavanis, K.P. Forecasting stock market short-term trends using a neuro-fuzzy based methodology. *Expert Syst. Appl.* **2009**, *36*, 10696–10707. [[CrossRef](#)]
6. Chollet, F. *Deep Learning with Python*, 1st ed.; Manning Publications Co.: Shelter Island, NY, USA, 2018.
7. Vanstone, B.; Finnie, G. An empirical methodology for developing stockmarket trading systems using artificial neural networks. *Expert Syst. Appl.* **2009**, *36*, 6668–6680. [[CrossRef](#)]
8. Zhong, X.; Enke, D. Forecasting daily stock market return using dimensionality reduction. *Expert Syst. Appl.* **2017**, *67*, 126–139. [[CrossRef](#)]
9. Fabozzi, F.J.; Focardi, S.M.; Kolm, P.N. *Quantitative Equity Investing: Techniques and Strategies*, 1st ed.; John Wiley & Sons: Hoboken, NJ, USA, 2010.
10. Cochrane, J.H. *New Facts in Finance*; Federal Reserve Bank of Chicago: Chicago, IL, USA, 1999; pp. 36–58.
11. Devendra, K. *Soft Computing: Techniques and Its Applications in Electrical Engineering*; Springer: Berlin, Germany, 2008.
12. Yin, S.; Jiang, Y.; Tian, Y.; Kaynak, O. A data-driven fuzzy information granulation approach for freight volume forecasting. *IEEE Trans. Ind. Electron.* **2016**, *64*, 1447–1456. [[CrossRef](#)]
13. Jiang, Y.; Yin, S. Recent advances in key-performance-indicator oriented prognosis and diagnosis with a matlab toolbox: Db-kit. *IEEE Trans. Ind. Inform.* **2018**, *15*, 2849–2858. [[CrossRef](#)]

14. Choudhary, A.; Upadhyay, K.; Tripathi, M. Soft computing applications in wind speed and power prediction for wind energy. In Proceedings of the 2012 IEEE Fifth Power India Conference, Murthal, India, 19–22 December 2012.
15. Engle, R.F. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econom. J. Econom. Soc.* **1982**, *50*, 987–1007. [[CrossRef](#)]
16. Atsalakis, G.S.; Valavanis, K.P. Surveying stock market forecasting techniques-Part I: Conventional methods. *J. Comput. Optim. Econ. Financ.* **2010**, *2*, 45–92.
17. De Luca, G.; Genton, M.G.; Loperfido, N. A skew-in-mean GARCH model for financial returns. In *Skew-Elliptical Distributions and Their Applications: A Journey Beyond Normality*; CRC/Chapman & Hall: Boca Raton, FL, USA, 2004.
18. Breiman, L. Statistical modeling: The two cultures. *Stat. Sci.* **2001**, *16*, 199–231. [[CrossRef](#)]
19. Gerlein, E.A.; McGinnity, M.; Belatreche, A.; Coleman, S. Evaluating machine learning classification for financial trading: An empirical approach. *Expert Syst. Appl.* **2016**, *54*, 193–207. [[CrossRef](#)]
20. Arnott, R.D.; Harvey, C.R.; Markowitz, H. A backtesting protocol in the era of machine learning. *J. Financ. Data Sci.* **2019**, *1*, 64–74. [[CrossRef](#)]
21. Di Matteo, T. Multi-scaling in finance. *Quant. Financ.* **2007**, *7*, 21–36. [[CrossRef](#)]
22. Yu, L.; Wang, S.; Lai, K.K. Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. *Energy Econ.* **2008**, *30*, 2623–2635. [[CrossRef](#)]
23. Atsalakis, G.S.; Valavanis, K.P. Surveying stock market forecasting techniques-Part II: Soft computing methods. *Expert Syst. Appl.* **2009**, *36*, 5932–5941. [[CrossRef](#)]
24. Bahrammirzaee, A. A comparative survey of artificial intelligence applications in finance: Artificial neural networks, expert system and hybrid intelligent systems. *Neural Comput. Appl.* **2010**, *19*, 1165–1195. [[CrossRef](#)]
25. De Luca, G.; Genton, M.G.; Loperfido, N. A Multivariate Skew-Garch Model. In *Advances in Econometrics: Econometric Analysis of Economic and Financial Time Series; Part A* (Special volume in honor of Robert Engle and Clive Granger, the 2003 Winners of the Nobel Prize in Economics); Terrell, D., Ed.; Elsevier: Oxford, UK, 2006.
26. Hassan, M.R.; Nath, B. Stock market forecasting using hidden Markov model: A new approach. In Proceedings of the 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), Warsaw, Poland, 8–10 September 2005.
27. Hassan, M.R. A combination of hidden Markov model and fuzzy model for stock market forecasting. *Neurocomputing* **2009**, *72*, 3439–3446. [[CrossRef](#)]
28. Hassan, M.R.; Nath, B.; Kirley, M. A fusion model of HMM, ANN and GA for stock market forecasting. *Expert Syst. Appl.* **2007**, *33*, 171–180. [[CrossRef](#)]
29. Gupta, A.; Dhingra, B. Stock market prediction using hidden markov models. In Proceedings of the 2012 Students Conference on Engineering and Systems, Allahabad, India, 16–18 March 2012.
30. Zhang, X.; Li, Y.; Wang, S.; Fang, B.; Philip, S.Y. Enhancing stock market prediction with extended coupled hidden Markov model over multi-sourced data. *Knowl. Inf. Syst.* **2018**, *61*, 1071–1090. [[CrossRef](#)]
31. Schmidt, M. Identifying speaker with support vector networks. In Proceedings of the Interface '96 Sydney, Sydney, Australia, 9–13 December 1996.
32. Vapnik, V. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer Science & Business Media: New York, NY, USA, 2013.
33. Kim, K. Financial time series forecasting using support vector machines. *Neurocomputing* **2003**, *55*, 307–319. [[CrossRef](#)]
34. Tay, F.E.; Cao, L. Application of support vector machines in financial time series forecasting. *Omega* **2001**, *29*, 309–317. [[CrossRef](#)]
35. Huang, W.; Nakamori, Y.; Wang, S. Forecasting stock market movement direction with support vector machine. *Comput. Oper. Res.* **2005**, *32*, 2513–2522. [[CrossRef](#)]
36. Henrique, B.M.; Sobreiro, V.A.; Kimura, H. Stock price prediction using support vector regression on daily and up to the minute prices. *J. Financ. Data Sci.* **2018**, *4*, 183–201. [[CrossRef](#)]
37. Rustam, Z.; Kintandani, P. Application of Support Vector Regression in Indonesian Stock Price Prediction with Feature Selection Using Particle Swarm Optimization. *Model. Simul. Eng.* **2019**, *2019*. [[CrossRef](#)]
38. Lee, M. Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Syst. Appl.* **2009**, *36*, 10896–10904. [[CrossRef](#)]

39. Zhang, G.; Patuwo, B.E.; Hu, M.Y. Forecasting with artificial neural networks: The state of the art. *Int. J.* **1998**, *14*, 35–62.
40. White, H. Economic prediction using neural networks: The case of IBM daily stock returns. In Proceedings of the IEEE 1988 International Conference on Neural Networks, San Diego, CA, USA, 24–27 July 1988.
41. Kimoto, T.; Asakawa, K.; Yoda, M.; Takeoka, M. Stock market prediction system with modular neural networks. In Proceedings of the IEEE International Joint Conference on Neural Networks, San Diego, CA, USA, 17–21 June 1990.
42. Qiu, M.; Song, Y. Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLoS ONE* **2016**, *11*, e0155133. [[CrossRef](#)]
43. Kara, Y.; Boyacioglu, M.A.; Baykan, Ö.K. 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.* **2011**, *38*, 5311–5319. [[CrossRef](#)]
44. Cao, Q.; Leggio, K.B.; Schniederjans, M.J. A comparison between Fama and French’s model and artificial neural networks in predicting the Chinese stock market. *Comput. Oper. Res.* **2005**, *32*, 2499–2512. [[CrossRef](#)]
45. Song, Y.; Zhou, Y.; Han, R. Neural networks for stock price prediction. *arXiv* **2018**, arXiv:1805.11317.
46. Lawrence, S.; Giles, C.L.; Tsoi, A.C. Lessons in neural network training: Overfitting may be harder than expected. In Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, Providence, Rhode Island, 27–31 July 1997.
47. Adebisi, A.A.; Adewumi, A.O.; Ayo, C.K. Comparison of ARIMA and artificial neural networks models for stock price prediction. *J. Appl. Math.* **2014**, *2014*, 1–7. [[CrossRef](#)]
48. Han, B. *Framelets and Wavelets: Algorithms, Analysis, and Applications*, 1st ed.; Birkhäuser: Basel, Switzerland, 2018.
49. Mallat, S. *Wavelet Tour of Signal Processing*, 3rd ed.; Academic: New York, NY, USA, 2008.
50. Chui, C.K. *An Introduction to Wavelets*, 1st ed.; Academic Press: San Diego, CA, USA, 2016.
51. Daubechies, I. *Ten Lectures on Wavelets*; Siam: Philadelphia, PA, USA, 1992.
52. Han, B.; Zhuang, X. Matrix extension with symmetry and its application to symmetric orthonormal multiwavelets. *Siam J. Math. Anal.* **2010**, *42*, 2297–2317. [[CrossRef](#)]
53. Zhuang, X. Digital affine shear transforms: Fast realization and applications in image/video processing. *Siam J. Imaging Sci.* **2016**, *9*, 1437–1466. [[CrossRef](#)]
54. Gurley, K.; Kareem, A. Applications of wavelet transforms in earthquake, wind and ocean engineering. *Eng. Struct.* **1999**, *21*, 149–167.
55. Gençay, R.; Selçuk, F.; Whitcher, B. Scaling properties of foreign exchange volatility. *Phys. A Stat. Mech. Appl.* **2001**, *289*, 249–266. [[CrossRef](#)]
56. Lahmiri, S. Wavelet low-and high-frequency components as features for predicting stock prices with backpropagation neural networks. *J. King Saud Univ. Comput. Inf. Sci.* **2014**, *26*, 218–227. [[CrossRef](#)]
57. Li, J.; Shi, Z.; Li, X. Genetic programming with wavelet-based indicators for financial forecasting. *Trans. Inst. Meas. Control* **2006**, *28*, 285–297. [[CrossRef](#)]
58. Hsieh, T.; Hsiao, H.; Yeh, W. Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm. *Appl. Soft Comput.* **2011**, *11*, 2510–2525. [[CrossRef](#)]
59. Wang, J.; Wang, J.; Zhang, Z.; Guo, S. Forecasting stock indices with back propagation neural network. *Expert Syst. Appl.* **2011**, *38*, 14346–14355. [[CrossRef](#)]
60. Jothimani, D.; Shankar, R.; Yadav, S.S. Discrete wavelet transform-based prediction of stock index: A study on National Stock Exchange Fifty index. *J. Financ. Manag. Anal.* **2015**, *28*, 35–49.
61. Chandar, S.K.; Sumathi, M.; Sivanandam, S. Prediction of stock market price using hybrid of wavelet transform and artificial neural network. *Indian J. Sci. Technol.* **2016**, *9*, 1–5. [[CrossRef](#)]
62. Khandelwal, I.; Adhikari, R.; Verma, G. Time series forecasting using hybrid ARIMA and ANN models based on DWT decomposition. *Procedia Comput. Sci.* **2015**, *48*, 173–179. [[CrossRef](#)]
63. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.; Tung, C.C.; Liu, H.H. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [[CrossRef](#)]

64. Huang, N.E.; Wu, Z. A review on Hilbert-Huang transform: Method and its applications to geophysical studies. *Rev. Geophys.* **2008**, *46*. [[CrossRef](#)]
65. Wei, L. A hybrid ANFIS model based on empirical mode decomposition for stock time series forecasting. *Appl. Soft Comput.* **2016**, *42*, 368–376. [[CrossRef](#)]
66. Drakakis, K. Empirical mode decomposition of financial data. *Int. Math. Forum* **2008**, *3*, 1191–1202.
67. Nava, N.; Di Matteo, T.; Aste, T. Financial time series forecasting using empirical mode decomposition and support vector regression. *Risks* **2018**, *6*, 7. [[CrossRef](#)]
68. Cheng, C.; Wei, L. A novel time-series model based on empirical mode decomposition for forecasting TAIEX. *Econ. Model.* **2014**, *36*, 136–141. [[CrossRef](#)]
69. Lin, C.; Chiu, S.; Lin, T. Empirical mode decomposition-based least squares support vector regression for foreign exchange rate forecasting. *Econ. Model.* **2012**, *29*, 2583–2590. [[CrossRef](#)]
70. Zeng, Q.; Qu, C. An approach for Baltic Dry Index analysis based on empirical mode decomposition. *Marit. Policy Manag.* **2014**, *41*, 224–240. [[CrossRef](#)]
71. Yu, H.; Liu, H. Improved stock market prediction by combining support vector machine and empirical mode decomposition. In Proceedings of the 2012 Fifth International Symposium on Computational Intelligence and Design, Hangzhou, China, 28–29 October 2012.
72. Peiro, A. Skewness in financial returns. *J. Bank. Financ.* **1999**, *23*, 847–862. [[CrossRef](#)]
73. Shcherbakov, M.V.; Brebels, A.; Shcherbakova, N.L.; Tyukov, A.P.; Janovsky, T.A.; Kamaev, V.A. A survey of forecast error measures. *World Appl. Sci. J.* **2013**, *24*, 171–176.
74. Willmott, C.J.; Matsuura, K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Clim. Res.* **2005**, *30*, 79–82. [[CrossRef](#)]
75. Sakamoto, Y.; Ishiguro, M.; Kitagawa, G. *Akaike Information Criterion Statistics*; KTK Scientific Publishers: Dordrecht, The Netherlands, 1986.
76. Martin, J.H.; Jurafsky, D. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*; Pearson/Prentice Hall: Upper Saddle River, NJ, USA, 2009.
77. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [[CrossRef](#)]
78. Madge, S.; Bhatt, S. *Predicting Stock Price Direction Using Support Vector Machines*; Independent Work Report Spring; Princeton University: Princeton, NJ, USA, 2015.
79. Sheela, K.G.; Deepa, S.N. Review on methods to fix number of hidden neurons in neural networks. *Math. Probl. Eng.* **2013**, 2013. [[CrossRef](#)]
80. Subasi, A. EEG signal classification using wavelet feature extraction and a mixture of expert model. *Expert Syst. Appl.* **2007**, *32*, 1084–1093. [[CrossRef](#)]
81. Balvers, R.; Wu, Y.; Gilliland, E. Mean reversion across national stock markets and parametric contrarian investment strategies. *J. Financ.* **2000**, *55*, 745–772. [[CrossRef](#)]
82. Fama, E.F.; French, K.R. Size, value, and momentum in international stock returns. *J. Financ. Econ.* **2012**, *105*, 457–472. [[CrossRef](#)]
83. Bai, L.; Yan, S.; Zheng, X.; Chen, B.M. Market turning points forecasting using wavelet analysis. *Phys. A Stat. Mech. Appl.* **2015**, *437*, 184–197. [[CrossRef](#)]

