A Comprehensive Study of Market Prediction from Efficient Market Hypothesis up to Late Intelligent Market Prediction Approaches

Amin Aminimehr1 · Ali Raoofi2 · Akbar Aminimehr3 · Amirhossein Aminimehr4

Accepted: 12 January 2022 / Published online: 14 June 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
This paper has scrutinized the process of testing market efficiency, data generation process and the feasibility of market prediction with a detailed, coherent and statistical approach. Furthermore, attempts are made to extract knowledge from S&P 500 market data with an emphasize on feature engineering. As such, different data representations are provided through different procedures, and their performance in knowledge extraction is discussed. Amongst the neural networks, Long Short-Term Memory has not been adequately experimented. LSTM, because of its intrinsic, considers the long-term and short-term memory in its computations. Thus, in this paper LSTM is further examined in return prediction and different preprocessing methods are tested to improve its accuracy. This study is conducted on market data during September-2000 to February-2021. In order to extend the amount of knowledge extracted from financial time series, and to select the best input features, the advantage of Principal Component Analyze, Random Forest, Wavelet and the LSTM’s own deep feature extraction procedure are taken, and 4 models are compiled. Subsequently, to validate the performance of the models, MAE, MSE, MAPE, CSP and CDCP are calculated. Results from Diebold Mariano test implied that although LSTM neural network has gained a lot of attention recently, it does not significantly perform better than the benchmark method in S&P 500 index return prediction. Yet, results from Wilcoxon signed rank test showed the significance of improvement in the predictions performed by the combination of Principal component analysis and LSTM.

Keywords Adaptive market hypothesis · Efficient market hypothesis · Financial forecasting · Data generation process · Stock returns

Extended author information available on the last page of the article
1 Introduction

1.1 Predictability of stock market

Predictability of stock market has been argued for a long time. Before the claim of Fama, there were different researches held by various scholars about the predictability of stock market, but 1965 was the milestone in the literature of this area in finance. In that year, Fama and Samuelson stated their own beliefs on the behavior of financial time series. Fama, for the first time, suggested the term “efficient market”, and stated that price time series in strong efficient markets follow a random walk process. At the same year, Samuelson suggested the martingale stochastic process instead of random walk for financial time series behavior. Besides the differences of their point of view, they both concur on unprofitability of predictions in efficient stock markets especially with the advantage of historical data; see Fama (1970) and Samuelson (2013). Accordingly, studying the evidence of efficient market hypothesis (EMH) to determine the degree of efficiency in different markets is highly recommended in prediction surveys.

Although many researchers have applied various methodologies with the aim of studying market efficiency; see Abeysekera (2001), Dias and Peters (2020), Dickinson and Muragu (1994), Dima and Mioloş (2009), Gordon and Rittenberg (1995) and Pele and Voineagu (2008) this investigation does not seem to be straightforward. Surveys held on EMH may have lacked considering the nonlinear behavior of data; see Basu (1977), Busse & Clifton Green (2002), Jensen (1978) and Rosenberg et al. (1985), as a result they may have falsely accepted the efficiency of market in different levels. In this regard, a research tested the random-walk hypothesis with volatility-based specification test. They found strong evidence for rejecting random walk hypothesis in weekly data of CRSP stock market. They also stated the evidence of long-term memory and predictability to some extends; see Doan and Lo (1988) and Lo (1991). Furthermore, Broock et al. (1996) developed another test called BDS with the aim of effectively studying independently and identically distribution (IID) in the behavior of data. This test could more effectively study financial time series and their behavior with a non-linear approach, as a result studying EMH in different researches became deeper.

Despite the developments in the approaches assessing market efficiency, there are still many discussions in this regard. From 1980, a group of researchers developed the idea of investors behavioral effects on market movements. This idea was first witnessed by Shiller (1981) while studying the movements of Standard & Poor (S&P) return after every time a new information was supplied in the market. He could observe excess unexplained volatility (by the literature of classic EMH) after each release of information. In this regard, in 2004, Lo explained a new definition for efficiency in financial markets with the term adaptive market hypothesis (AMH). In this definition, as investors behavior is not considered rational but a combination of complicated psychological factors; see Barber and Odean (2000) and Shleifer (2000), there may be periods of time that market spends on adapting to new circumstances;
see Lo (2004). This branch of literature, nowadays, is also continued with researches on investor’s sentiments.

After the developed notion by Lo, approaches upon studying the efficiency of market changed. It seems that by the developments in the machine learning knowledge extraction algorithms, or the availability of quantitative and qualitative data like what is available on social media, attitudes towards the level of market efficiency has changed. Accordingly, the efficiency of those markets that were confirmed may not still be confirmed. As a result, there is a rich literature devoted to the area of finance which takes use of mathematics and statistics with the aim of modeling and predicting financial time series. These methods vary from early statistic and econometric methods up to the late machine learning (ML) and evolutionary based optimization techniques; see e.g. Kanas and Yannopoulos (2001), Kim (2003, 2006), Park et al. (2010), Rather et al. (2015) and Han et al. (2020).

There are many studies aiming at modeling financial time series with different frontier methods, however it should be kept in mind that financial predictions should be performed consciously and cautiously by the pre-studies conducted on data. Thus, in this research attempts are made to predict S&P 500 index with the latest advanced Deep Learning (DL) methods. The novelty of this research is that it’s emphasize is on the pre-estimations of data and the process of feature engineering.

The remainder of the paper is organized as follows. In Sect. 1.2 the literature of financial predictions using machine learning methods and their benefits against classic methods is summarized. Section 2 explains the materials including the statistical tests and methods used in the study. Section 3 is related to the studies on the characteristics of the used data. Section 4 explains the methodology and the conducted experiments for validating different feature engineering methods. Section 5 is devoted to explaining the metrics and statistical test used for validating the composed models. Section 6 exposes the results and discussions. Finally, Sect. 7 concludes the study.

1.2 Prediction models

Artificial Neural Networks (ANN) are a class of complex computing models inspired from human’s neural network system, and are a class of ML algorithms. As human’s brain can solve many complex problems by learning the underlying relations and patterns between variables, ANN’s are expected to solve many problems that earlier methods couldn’t. This simulation of the biological process of thinking and solving has been used in different subjects for about 80 years; see Adya and Collopy (1998), McCulloch and Pitts (1943) and Tsaih et al. (1998). Besides, ANNs have been even successful in predicting cycles that included crisis periods with an accuracy margin error of bellow 5% Yavuz et al. (2015).

Although many believe that ANNs unconditionally can perform better than conventional statistical methods in modeling financial time series, inconsistent results of recent literatures; see McAleer and Medeiros (2011) in using them on financial markets has led to controversy amongst scholars. Inconsistent results in the success of ANN’s against earlier methods is due to various reasons such as inappropriate model implementation (for example model selection and hyperparameter selection),
and neglecting to study the characteristics of data. Nowadays, with the developments of artificial intelligence (AI) in Deep Neural Network’s and model implementing techniques, hopes are increasing to solve the first problem. Keeping in mind that the answer to the second problem lies in studying the data generating process (DGP). It seems that in some cases when there is not a linearity and non-linearity specification on data, controversial results occur. For example, in some cases while ANN’s are struggling with unnecessary complexity of data, linear nature in the relation of features and target value, makes linear classic models more effective in modeling them. This problem can be prevented by an adequate study on linearity and non-linearity characteristic of data which results in a more coherent model selection. Eğrioğlu and Fildes (2020) have considered this specification through an input significance test in their research.

After all, it is obvious that conventional statistical methods have many limitations in studying financial data. Problems such as high number of computations, losing degree of freedom, facing redundant and omitted variables, and multicollinearity between input features. In financial predictions, due to the high number of variables, there is a need for a clear and strong feature selection approach to avoid the problems mentioned above. Unfortunately, classic models lack an effective procedure of feature selection too. consequently, scholars used to select features either by chance, or by calculating linear correlation and maybe earlier literatures recommendations. For example, Hajizadeh et al. (2012) mentioned in their paper that the exogenous input variables for their hybrid model is selected from the recommendations of some earlier researches. Besides, stock market is a dynamic system that may change its dependency pattern through time and in different economic situations, therefore, there must be a suitable feature engineering approach for a better prediction.

Moreover, although GARCH family models are known for their non-linear modeling capability, they are designed for modeling volatility instead of return and cannot sufficiently capture long term dependencies in data. Therefore, classic models seem to be insufficient at modeling financial time series. As a result, many scholars have inevitably switched to more complicated methods which can better capture long-term memory.

Fortunately, along with the developments of big data and data science, new procedures are suggested to deal with the problems that earlier methods couldn’t. For example, there are various methods proposed for feature engineering before deploying prediction models. Within feature engineering approaches, there are some methodologies concentrating on the curse of dimensionality. Feature extraction and feature selection are two branches of feature engineering with the aim of dimensionality reduction. The main difference between these two approaches is that in feature selection, subsets of the original features are provided by the method, but in feature extraction brand new feature are created regarding to the initial features. Random Forest (RF); see Tin Kam (1995) and Principal Component Analysis (PCA); see Gastpar et al. (2006) respectively are well known examples of each feature engineering methods introduced. In addition, lately, intelligent feature engineering is performed by deep neural networks like Recurrent Neural Network’s (RNN) and Convolutional Neural Networks (CNN); see LeCun and Bengio (1998). In another words, deep neural networks have provided a feature extraction procedure.
in addition to their mapping functionality through the training process. Conventional simple neural networks seem to not provide this feature extraction process, as a result many scholars have switched to CNN’s and RNN’s like Long Short-Term Memory (LSTM); see Hochreiter and Schmidhuber (1997) networks for their studies.

Literature of finance and deep learning have had many contributions in recent years. Various methods of DL models have penetrated through diversified subfields of finance such as algorithmic trading, risk assessment, fraud detection, portfolio management, asset pricing, financial statement analysis, text mining, behavioral finance, etc.; see Ozbayoglu et al. (2020). In Table 1 there is a list provided of recent literatures of machine learning and more specifically DL methods used in finance.

With the advanced developments in programming languages especially python and its specific libraries of data science such as Sikitlearn, Pytorch, Tensorflow, Scipy etc., running various types of models have become extremely convenient. Hence, the main concern of this paper is not to seek for a new model or to create one, but to manage a unique methodology that completes the earlier researches of financial forecasts. The main aim of this paper is twofold. First is to sufficiently study the characteristics of financial time series and then better incorporate explanatory variables in predictions. The stated criteria are lacking in many researches dedicated to stock prediction, however studying them results in more reasonable, accurate and reliable predictions.

In this research, first, investigations are conducted to gain a primary perception on the characteristics of the studied data which gives an insight on the process of data generation. Through this step EMH is accordingly studied on the S&P 500 data. Next, attempts are made to improve the accuracy of a specific ML model (LSTM), experiment by experiment, by modifying the procedure of feature selection, feature extraction and denoising. Indeed, the model is primarily fed with a variety of input variables and then features are extracted with different methods. The experiments in this paper are as follows:

1. PCA-LSTM
2. RF-LSTM
3. Deep LSTM
4. Wavelet-Deep LSTM

The flowchart in Fig. 1 shows the overview of the second part of the methodology.

2 Materials and Methods

2.1 Statistical Tests and Feature Generating Methods

Implementing prediction models needs to be in accordance to the characteristics of data. Therefore, a comprehensive study on the characteristics of data must be held by descriptive statistics and statistical tests before model implementation. Various tests have been developed in order to study various aspects of time series data. Within different tests designed for time series, Augment Dicky fuller is designed for testing stationarity in data. From the other side, ARCH-LM test; see Bollerslev (1986) is
| Topic of study                  | Date      | Article                        | Method         | Variables                                                                 | Study period       | Results                                                                                                                                                                                                 |
|--------------------------------|-----------|-------------------------------|----------------|---------------------------------------------------------------------------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bankruptcy and crisis forecasting | 2018      | Chatzis et al. (2018)         | SVM, RF, ANN, etc | High number of variables including various stock indices, exchange rates, etc | 1996–2017          | Deep learning methods offer capabilities that can excitingly predict out of sample crisis events as an early warning system                                                                         |
|                                | 2018      | Kvatame et al. (2018)         | CNN            | Customer’s transaction data such as checking account, saving account, checking transactions, etc | 2012–2016          | Combination of Random Forest classifier and CNN can achieve better forecasting accuracy in predicting mortgage default                                                                             |
| Fraud detection                | 2018      | Jurgovsky et al. (2018)       | LSTM           | Time, transaction type, Amount of transaction, merchant group, bank, etc    | 2015–2015          | LSTM gained enough accuracy in predicting offline transactions fraud detection, yet combination of LSTM and Random Forest may be better in summary                                                      |
| Portfolio management           | 2020      | Ta et al. (2020)              | LSTM           | Multivariate—open prices, Close Prices, Highest prices, Lowest prices, Trading volume—500 large capital stocks of US Stock Market | 2008–2018          | LSTM performs more accurate forecasts than Linear Regression and Support Vector Machine to enhance the process of portfolio optimization                                                                  |
|                                | 2020      | Wang et al. (2020)            | LSTM           | Multivariate—Adjusted open prices, Close Prices, Highest prices, Lowest prices, Trading volume -UK Stock Exchange 100 Index | 1994–2019          | LSTM networks can enhance the result of portfolio optimization                                                                                                                                        |
|                                | 2019      | Vukovic et al. (2020)         | ANN            | Multivariate—representation of low risk and high-risk assets—fixed income assets (EU market) | 2004–2017          | Neural networks are successful in predicting Sharpe- ratio dynamics in the future                                                                                                                     |
| Text mining                    | 2020      | Bari and Agah (2018)          | LSTM, GRU, etc | Text features mainly extracted from tweets and stock quotes                | 2016–2016          | Several models in terms of event clustering, text extraction and regression are proposed and compared them with Sharpe ratio                                                                            |
|                                | 2020      | Jin et al. (2020)             | CNN-Attention  | Sentiment index, Open, High, Low, trend of close price, Volume            | 2013–2018          | LSTM as a base learner combined with CNN for sentiment analysis could achieve a good prediction of close price, a good classification of rise and falls of price, and lower offset of time in predictions |
| Topic of study                | Date   | Article                  | Method                        | Variables                                                                 | Study period | Results                                                                                                                                 |
|------------------------------|--------|--------------------------|-------------------------------|--------------------------------------------------------------------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Volatility and return prediction | 2020   | Bucci (2020)             | LSTM                          | Multivariate—logarithm of realized volatility—set of economic, financial and market factors | 1950–2017   | Results showed that long term dependence detection models like LSTM can outperform classic econometric and other neural network in volatility prediction |
|                              | 2019   | Zhong and Enke (2019)    | ANN                           | Multivariate—Market data and 60 fundamental variables                   | 2003–2013   | ANN-PCA can acceptably predict price return by dimensionality reduction which leads to trading strategies with higher profit than market benchmark         |
|                              | 2018   | Kim and Won (2018)       | LSTM-GARCH                    | Multivariate—Gold price, Oil price, CB interest rate, KTB interest rate—KOSPI 200 Index | 2001–2017   | Combining LSTM with double or more GARCH models significantly improves predictions accuracy                                                   |
|                              | 2017   | Yao, et al. (2017)       | (ANN) enhanced two-component (NNE2C) | Univariate—Exchange rate—Oricode Inc                                     | 2009–2015   | Their proposed model with 2 enhancement components of long-term and short-term outperforms almost every earlier GARCH methods                   |
|                              | 2017   | Kristjanpoller and Hernández P (2017) | ANN-GARCH                    | Multivariate—Commodity prices gold, silver and copper, explanatory variables are exchange rates, oil price return, stock market index returns etc | 1999–2014   | Neural networks did improve the prediction accuracy of GARCH family models within different time step predictions                             |
|                              | 2015   | Kristjanpoller and Minutolo (2015) | ANN-GARCH                    | Univariate—GARCH factors—Brazilian BOVESPA, Chilean IPSA, Mexican IPyC    | 2000–2012   | Improvement of MAPE while using ANN’s in comparison to GARCH model                                                                     |
|                              | 2012   | Hajizadeh, et al. (2012) | EGARCH-ANN                   | Multivariate—lags of different explanatory variables—NASDAQ, Dow Jones and target lags (S&P 500) | 1998–2009   | Their hybrid model could outperform the simple EGARCH (3,3), but it had weakness in capturing violent volatility                           |
| Topic of study | Date       | Article                        | Method      | Variables                                                                 | Study period  | Results                                                                 |
|---------------|------------|--------------------------------|-------------|--------------------------------------------------------------------------|---------------|--------------------------------------------------------------------------|
|               | 2009       | Bildirici and Ersin (2009)     | ANN-APGARCH | Univariate—Impact of domestic and international good and bad news—ISE National-100 Index from Central Bank of Turkey Database | 1987–2008     | By supporting the GJR (threshold) GARCH with ANN, forecasting accuracy improved |
|               | 2009       | Wang (2009)                     | GJR GARCH-ANN | Multivariate—primary Blach-Scholes variables—Option price (target value)—Taiwan stock index | 2005–2007     | Grey GJR GARCH model can even further improve the forecasting ability of hybrid GJR GARCH model with ANNs |
|               | 2007       | Roh (2007)                      | ANN-conventional statistical methods | Univariate—KOSPI 200 Index                                             | —             | Results implied that ANNs can enhance the prediction power of EGARCH methods and their direction prediction accuracy |
designed to test the presence of heteroskedasticity. Furthermore, ARCH correlogram can be used for testing serial conditional variance correlation in time series.

In term of studying the Data Generation process of time series, BDS test can be applied on the studied data in three steps. BDS test’s null hypothesis is that the tested time series is independent and identically distributed. This test can be applied on residual series of an estimation too. If residual series of a linear estimation is tested by BDS, rejecting the null hypothesis will imply the presence of nonlinear characteristics in data. In another word, it demonstrates that the defined knowledge in the series by a linear estimation has not explain all the knowledge in the series, and there is an unexplained non-linear knowledge left in the residuals. Furthermore, if the residual series of a non-linear model is tested by BDS, rejecting the null hypothesis will imply the presence of chaotic knowledge in data. In another words, it means that by explaining non-linear characteristics of data there is still a chaotic information left unexplained in the residuals. Figure 2 has depicted this procedure in a more perceptive form.

The BDS test statistics is defined as bellow:

\[
W_{m,T}(\epsilon) = \frac{\sqrt{T} \left[ C_{m,T}(\epsilon) - C_{m,T}(\epsilon)^n \right]}{\sigma_{m,T}(\epsilon)}
\]

(1)

In this test, sets of pared points are selected and their differences are calculated. A time series is IID if the probability of the differences are smaller or equal to \( \epsilon \). Usually, the \( \epsilon \) is a multiple of the standard deviation of the studied time series and chosen by the analyst. The mentioned probability is denoted by \( C_T(\epsilon) \).

By observing specific properties and characteristics in return series, many approaches have been introduced in recent literatures to define them. As return series are dominated by intense volatilities which makes them complex and hard to model, researchers developed the idea of modeling using historical standard deviation as the explanatory factor; see Engle (1982). Therefore, various models of ARCH and GARCH were proposed for heteroscedasticity modeling. In continue, models such as EGARCH, EWGARCHG, GJR-GARCH, APGARCH, FIGARCH were additionally introduced to deal with asymmetry, long-term memory, etc. see Baillie et al. (1996), Bollerslev (1986), Glosten et al. (1993) and Nelson (1991).

Furthermore, in 1989 Hamilton introduced the Expectation Maximization (EM) based method for defining the clustered behavior; see Mandelbrot (1997) of time

![Fig. 1](https://example.com/fig1.png) Experiments regarding to pre-processing methods (color used)
series data. He believed that the distribution of time series data may be generated from 2 or more distinct distributions. These distributions were specifically called regimes. Indeed, it may be the discrete shift in regimes with different governing parameters that causes different cycles in time series. As the data generation process of return is considered to be from more than one distribution, and the governing state variable is latent, expectation maximization algorithm was recommended to estimate the parameters. On the contrary to Quandt’s model; see Quandt (1972), Hamilton (1988) believed that the shift in regimes is controlled by an unobservable state variable which follows a first order Markov chain. As a result, two or more (depending on the number of detected regimes) separate dynamic models were combined through a state variable ($s_t$) to imitate more than one business cycles. In summary Hamilton combined more than one estimation into a unit polynomial formula to better characterize the dynamics of financial time series:

$$y_t = y_{t-1} \beta_{s_t} + z_t \sigma + \epsilon_t,$$

$$u_t | s_t \sim N(0, \Sigma_{s_t}).$$

In this formula $y_{t-1}$ is a regime dependent and $z_t$ is a regime independent variable and by assuming that there are two regimes, the transition matrix will be defined as follows:

$$P = \begin{bmatrix} P_{11} & P_{21} \\ P_{12} & P_{22} \end{bmatrix} \text{ with } \sum_{j=1}^{2} P_{kj} = 1 \text{ and } P_{kj} \geq 0 \forall k, j \in \{1, 2\}$$

By the developments of various methods for defining various features of time series, lack of a feature selection method was highlighted more than ever before. It was the time for developing feature selection and feature extraction approaches to give a better chance to estimation methods to incorporate the explanatory variables. Furthermore, stock market time series are very likely to be non-linear, as a result artificial intelligence estimation tool can be much more helpful.
2.2 LSTM Networks

ML as a branch of AI and data mining has provided various algorithms for solving different problems. They are based on experience and examination on data. These algorithms are divided into many branches such as supervised learning, unsupervised learning, reinforcement learning, etc.

DL, support vector machine (SVM), k-nearest neighbors (kNN), decision trees (DT) and RF are various algorithms of ML. Within these models, DL which can be implemented by ANNs possess the advantage of unsupervised feature learning, mechanisms to avoid overfitting, and a fast and robust fitting on big data. Through recent years DL has been used in various topics, such as audio, image and video in order to perform classification and prediction; see Chai, et al. (2019). Even though DL was first applied in computer science, its applications is now vastly used in medicine, neuroscience, physics, astronomy, finance, banking, and operation management. This is because of its various practical achievements in solving human’s problems; see Huanget al. (2020) and Li and Sim (2013). The prevalence of ANN and DL is because of their ability in fitting non-linear relationships, which is useful in solving practical real-world problems; see Ko and Lin (2006, 2008). This feature was noted by financial researchers because of the emphasize they had on non-linear models.

LSTM networks are a class of DL methods. They have the ability to automatically extract features from the data and were introduced by Hochreiter and Schmidhuber (1997) as an improved method to learn sequential patterns rather than simples RNNs. In a research by Graves and Schmidhuber (2005) showed that LSTM networks are better at memorizing long term knowledge, and less often face the problem of vanishing gradient\(^1\) comparing to simple RNNs; see Fischer and Krauss (2018) and LeCun et al. (2015). Sang and Di Pierro (2019) showed that LSTM can be effective in improving the performance of trading algorithms in financial data.

The structure of LSTM network is shown in Fig. 3. LSTM networks have a simple but effective structure. Each cell individually is a system performing a precise function. Ignoring the inner sub-functions of LSTM block, the main function is to get three inputs simultaneously.

1. The input vector (input from recent time step)
2. Memory (cell state) passed from previous time step
3. Output of previous layer

and finally, to pass two information elements simultaneously.

1. Memory (cell state) of the current block
2. Output vector of the current block.

---

\(^1\) This occurs when weights of a neural network stops updating during training process, it may be due to various reasons such as very deep architecture in layers, selecting inappropriate activation function, etc. There are several ways to avoid this problem such as selecting alternative node structure, decreasing number of layers, using activation functions with lower saturation property.
Inside an LSTM cell there are three gates which form the recurrent gate: 1. The input gate, 2. Forget gate, and 3. Output gate. The input gate decides which data can be added into the cell state, the output gate decides which data from the cell state can be used as output, and the forget gate decides which data should be dismissed from cell state. In summary, the cell state is the memory of each LSTM cell and the feature extraction capability of LSTM is due to the presence of this state.

The described procedure inside each LSTM cell is according to specific sub-functions described in the following formula.

\[ f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \]  
\[ i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \]  
\[ O_t = \sigma(W_o \cdot [h_{t-1}, \tilde{f}_t] + b_o) \]  
\[ c_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \]  
\[ g_t = \tanh(x_tU^g + h_{t-1}W^g) \]  
\[ h_t = \tanh(c_t) \cdot 0 \]

With the developments of deep learning algorithms, scholars started feeding large number of features into their models and left the feature extraction process to the network itself. Besides, many feature extraction procedures are developed for dimensionality reduction which are implemented separate from the training process of classification and regression. In the literature of finance there is not an enough discrimination made between the ability of feature extraction in deep learning process and alternative tools, so this paper has additionally aimed at making a comparison between these methods.

### 2.3 Principal Component Analyze (PCA)

Principal component analyze is a method to reduce data dimensionality, and is within the class of unsupervised feature extraction methods. PCA in many ways forms the basis for multivariate data analysis and is a methodology through which an optimum number of series are generated to include the maximum amount of knowledge. In another words, through a PCA, the maximum amount of data variance is saved with the least possible amount of data series. In an overview PCA provides various functions such as simplification, data reduction, modeling, outlier detection, variable selection, classification, prediction and unmixing. Principal Component Analysis generates new features which are the linear combination of the initial features and has a procedure of matrix diagonalization through itself. PCA has proved to be enhancive for complex spaces, and various reasearchers have used this technique for denosing and dimensionality reduction in high dimensional data; see Murali et al. (2012). It has been shown that PCA has positive impact in the classification algorithm of intrusion detection; see Keerthi Vasan and Surendiran (2016) and it has also been enhancive in applications of volatility prediction too; see Kristjanpoller R and Hernández P.
Wold et al. (1987) recommends pre-scaling the data to unit variance which generally causes a better performance of knowledge retainement through PCA process.

### 2.4 Random Forest

RF is a class of ML algorithms, and is an improved method of decision trees; see Tin Kam (1995). This method has enhanced the overfitting problem of decision trees by creating multiple shallow decision tree architectures. RFs mainly consist of a number of decision trees, each of them built over a random extraction of the observations from the dataset and a random extraction of the features. Not every tree hosts all the features or all the observations, and this guarantees that the trees are de-correlated and therefore less prone to over-fitting. Each tree is also a sequence of yes–no questions based on a single or combination of features. At each node (here at each question), the tree divides the dataset into 2 buckets, each of them hosting observations that are more similar among themselves and different from the ones in the other bucket. Therefore, the importance of each feature is derived from how pure each of the buckets are.

RF seems to be enhancive in problem solving because they provide in general a good predictive performance, low chance of overfitting, and easy interpretability. This interpretability is given by the fact that it is straightforward to derive the importance of each variable on the tree decision. Feature selection using RF comes under the category of embedded methods. Embedded methods combine the qualities of filter and wrapper methods. They are implemented by algorithms that have their own built-in feature selection methods.

Random forest has found applications in financial researches. Researchers like Tyralis and Papacharalampous (2017) compared two sets of time series in their one step ahead prediction model, with the aim of introducing optimum set of predictor...
variables using RF. A Research by Jurgovksy et al. (2018) also used RF to select the most contribution features in their fraud detection system and found it enhancive while supporting their DL model.

2.5 Wavelet transforms

Wavelet transforms are vastly used by scholars in order to denoise various types of data. The wavelet approach, in addition to frequency decomposition has the ability to decompose within various time intervals. This capability is because of wavelets finite behavior which is further suggested in comparison with Furrier transform in analyzing time series. In a wavelet decomposing procedure the initial signal is broken into high level and low-level coefficients which can reconstruct the original signal. In the procedure of reconstructing the signals some of the high-level coefficients can be taken away to smoothen the reconstructed signal. There are mainly two types of wavelets known as discrete wavelet and continuous wavelet. Continuous wavelets because of extremely large number of computations can be computationally expensive to use. As a result, discrete wavelet is used more often. The procedure of discrete wavelet decomposition is described in the following formula:

\[
d_{j,n} = Wf(2^n \cdot n') = f(t, \psi_{j,n}(t)),
\]

\[
2^{-j} \int_{-\infty}^{+\infty} f(t) \cdot \psi(2^{-j}t - n) dt
\]

\[
\psi_{j,n} = 2^{-j/4} \psi(2^{j/2} - n).
\]

Wavelets can help machine learning algorithms to face a simpler form of data fed into, in hope of a more accuracy in training and prediction. As financial time series are extremely noisy due to many factors affecting them such as human interactions, scholars have lately started using wavelet in financial predictions too. This decomposition approach was used by Sharif et al. (2020) to analyze the connection between recent Corona virus spread with oil price volatility shock. Chang et al. (2019) used wavelet transform to stabilize their features in their hybrid LSTM model to predict Electricity price. Lu (2010) used wavelets advantage to denoise the features fed to their neural network model that predicts stock price. Raoofi and Mohammadi (2018) used wavelet in order to denoise Tehran Stock Exchanges Index data for a better forecast in their ANFIS. Li and Tang (2020) also proposed a model combining wavelet and Gated Recurrent Unit (GRU) network to predict S&P 500 price, and their model had the least error in prediction comparing with ARIMA and multi-lag SVR. Finally, in research held by Aminimehr et al. (2021), wavelets ability to denoise Tehran Stock Exchange index data was denoted as a method that improves prediction accuracy.
3 Data and Variables

In this research, the study period spans from September-2000 to February-2021. The research sample is extracted from yahoo finance and is one of the most prominent indices; S&P 500. This index is a capitalization-weighted index and the 10 largest companies in it account for 26% of the market capitalization of the index. Currently ten largest companies in this index, in order of weighting, are Apple Inc, Microsoft, Amazon.com, Alphabet Inc, Facebook, Johnson & Johnson, Berkshire Hathaway, Visa Inc., Procter & Gamble and JPMorgan Chase, respectively. Figure 4 shows the Close price of S&P 500 index during the studied period.

In order to have an insight on the studied data, descriptive statistics of the initial variables used in this study including Jarque–Bera test for testing normality is obtained. Results from the normality test rejects the null hypothesis which indicates that the distribution of the features are not normal by 95% confidence interval.

Predicting close price is the main aim of many financial researches, but in regression estimation, data must be stationary to avoid spurious estimations. As a result, unit root test is applied on close price in different levels to study stationarity. Results of the stationary test states the presence of trend component in close price. Summary of stationary test results are stated in Table 2. In order to completely examine non-stationarity by considering structural breaks in close price, unit root with break test is also applied after the results of Augment Dicky Fuller significantly accepted presence of unit root. To estimate the unit root with break test, Schwarz criterion is used for selecting optimum lags. Finally, all tests verify the presence of unit root in close price data. As a result, the target value of this research is switched to daily return which is significantly stationary with 99% confidence interval.

It should be kept in mind that some recent literatures of using AI methods on financial predictions have mentioned that these methods do not have any parametric presumptions; see Hajizadeh et al. (2012) and Qi (1996) and have neglected to mention non-stationarity in price data. Besides, although, Wang et al. (2020) has also stated in his paper that AI methods can deal with non-stationary time series better than conventional statistical methods, there is no strong examinations supporting such claim in the literature. Rather this is an exaggerated reliability on AI methods. Therefore, many of the researchers prefer to use return value instead of price value as the prediction target. The problem of predictions on non-stationary time series is the unbounded and unscaled characteristics of them, while prediction models may have never been trained for. Furthermore, applications of return prediction can directly be used in commercial fields such as algo-trading and portfolio optimization, while application of price prediction is not much reliable there.

Figure 5 shows the 1-day return of S&P 500 index and its histogram plot during the studied period. As mentioned in Table 3, results from Jarque Bera test, kurtosis and skewness implies that the distribution of the returns is not normal, heterogenous, and asymmetry. Furthermore, result from ARCH-LM test and ARCH correlogram Q-statistics significantly implies the presence of heteroscedasticity and serial correlation in S&P 500. The descriptive statistics of the return series is reported in Table 4.
To effectively study the linearity and non-linearity of data an Auto Regressive Moving Average (ARMA) model is estimated on the target value. If the residuals of the linear estimation are not IID then the DGP is more likely to be nonlinear or chaotic, and should be estimated by a non-linear estimator. Respectively, if the residuals of a non-linear estimation are not IID, then the DGP is likely to be chaotic.

In order to investigate the EMH and the DGP of S&P 500, BDS test results are obtained in 3 stages. The BDS test results implies that the return of S&P 500 is not IID. Result of BDS test on residual series of a linear estimation of return series rejects linear characteristics in S&P 500. Finally, results of BDS test on residual series of a non-linear estimation rejects non-linear characteristics in S&P 500. This investigation shows that financial time series cannot be completely explained with nonlinear estimations and follows a chaotic behavior. Tables 5, 6 and 7 have reported the result of the BDS test in the 3 defined stages.

### Table 2: Results for stationary tests

| S&P 500 | Augment dicky fuller (Trend & Intercept) | Augment dicky fuller (Intercept) | Augment dicky fuller (None) | Unit root with Break |
|---------|-----------------------------------------|---------------------------------|---------------------------|---------------------|
| Close price | $P$ value 0.526 | 0.989 | 0.968 | 0.213 |
| 1 day return | 0.000 | – | – | – |

To effectively study the linearity and non-linearity of data an Auto Regressive Moving Average (ARMA) model is estimated on the target value. If the residuals of the linear estimation are not IID then the DGP is more likely to be nonlinear or chaotic, and should be estimated by a non-linear estimator. Respectively, if the residuals of a non-linear estimation are not IID, then the DGP is likely to be chaotic.

In order to investigate the EMH and the DGP of S&P 500, BDS test results are obtained in 3 stages. The BDS test results implies that the return of S&P 500 is not IID. Result of BDS test on residual series of a linear estimation of return series rejects linear characteristics in S&P 500. Finally, results of BDS test on residual series of a non-linear estimation rejects non-linear characteristics in S&P 500. This investigation shows that financial time series cannot be completely explained with nonlinear estimations and follows a chaotic behavior. Tables 5, 6 and 7 have reported the result of the BDS test in the 3 defined stages.
In summary, to adequately and consciously implement a prediction model and to feed the models with appropriate explanatory variables, there must be variables included that have proven to be enhancing in explaining each of the characteristics detected in S&P 500 return. Characteristics such as non-linearity, asymmetry, heterogeneity, heteroscedasticity (clustered volatility) etc.

### 4 Methodology and Preliminary Analysis

The methodology in this research is first devoted to studying the efficiency of market by studying S&P 500 market data. Then, the research is continued to determine if the EMH is rejected, how can different advanced methods imitate the special detected characteristics of return series. To do so, different applications are implemented to compare the impact of different preprocessing methods in training neural networks. Finally, estimations and calculations are conducted to verify the difference between

| S&P 500     | Number of observations | Mean | Standard Deviation | Skewness | Kurtosis | Jarque–Bera |
|-------------|------------------------|------|--------------------|----------|----------|-------------|
| Close price | 5030                   | 1632.41 | 649.83           | 0.944    | −0.219   | 758.72      |
| Open price  | 5030                   | 1632.24 | 649.77           | 0.945    | −0.218   | 759.38      |
| Low price   | 5030                   | 1622.02 | 647.55           | 0.939    | −0.227   | 750.27      |
| High price  | 5030                   | 1641.57 | 651.60           | 0.950    | −0.212   | 766.96      |
| Volume      | 5030                   | 3,235,973,204.77 | 1,487,695,035.87 | 0.679    | 1.009    | 600.34      |

Fig. 5 S&P 500 index 1 day return–Red slider shows the major financial booms (color used)
the performance of the studied prediction methods. Figure 6 shows the methodology in detail.

The data in this study is split into three parts. First the training dataset which covers from 01/09/2000 up to 03/10/2018. This is the dataset by which the network is trained and weights are adjusted. The training data has covered the global financial crisis of 2008. Second is the validation dataset which covers from 04/10/2018 up to 11/09/2020. This set of data is used to validate the network after each update in weights. This set of data is used to halt the training process after the weights stop improving the accuracy of predictions. This set covers the duration of the recession caused by COVID-19 pandemic. Finally, the test data covers the period from 12/09/2020 up to 25/02/2021. This dataset is not used for halting the training process, so it is an absolutely unseen situation that is used to examine the prediction of the trained models. The problem of comparing prediction accuracy by validation data is that it has contributed to stop the training process on the best weights, as a result the reported accuracy may not have enough generalization.

4.1 Feature Generation

By observing various properties in the target value which manifests the fat tailed and clustered behavior of them, there are different features generated from the initial series in order to better imitate this characteristic in data. Thus, first, 24-time steps of GARCH and EGARCH are obtained as explanatory variables. It is not attempted to include EWGARCH variables because it seems that LSTM can extract a representation of EGARCH through its feature extraction which imitates EWGARCH already. Second, Average True Range (ATR) which is a volatility indicator and was first time introduced by J. Welles Wilder Jr is generated. ATR is a technical indicator which measures volatility through a trend of price and can give an insight of the strength or weakness of a trend. ATR is calculated by the following formula:

\[ \text{TR} = \text{Max}[(\text{High} - \text{Low}), \text{Abs}(\text{High} - \text{Close}), \text{Abs}(\text{Low} - \text{Close})] \]  

\[ \text{ATR} = \left( \frac{1}{n} \right) \sum_{i=1}^{n} TR_i. \]  

Third, a group of custom crafted dummy variables are included which gives information about the first lag of target value. The logic behind its generation is to

| Table 4 | Descriptive statistics for target value |
|---------|----------------------------------------|
| S&P 500 | Number of observations | Mean | Standard Deviation | Skewness | Kurtosis | Jarque–Bera |
| 1 day return | 5029 | 0.000 | 0.0125 | −0.148 | 11.219 | 26,394.04 |

 Springer
classify the data by their distance from the mean of the training with advantage of hard thresholding.

On the other side, Hamilton’s first order Markov switching model on the return series is estimated. Experiments implies that 2-regimes can better minimize the negative log-likelihood in comparison with up to 5 regimes. Significant regimes of different variance are observed in this model. As a result, the output of Hamilton’s model defining the marginal probability of the regime of the next day’s target value is also included in the explanatory variables of the model. Before adding this variable, they are passed through a hard thresholding function to produce zeroes and ones. This is done in order to prevent extreme complexity of the model. In summary, this model is estimated to make the outputs of LSTM network more flexible and sensitive to switches in regimes.

In summary, since the novelty of this study is verifying the strength of different pre-processing and feature engineering methods, each model is fed with a number of

| Table 5 | Results for BDS test on 1 day return S&P 500 |
|-----------------|---------------------|---------------------|---------------------|
| BDS test (return S&P 500) | Dimension | BDS statistics | P value |
| Epsilon=0.3 | 2 | 15.752 | 0.000 |
| | 3 | 24.364 | 0.000 |
| | 4 | 32.920 | 0.000 |
| | 5 | 43.289 | 0.000 |
| | 6 | 57.588 | 0.000 |

| Table 6 | Results for BDS test on the residuals of the linear estimation on 1 day return S&P 500 |
|-----------------|---------------------|---------------------|---------------------|
| BDS test (Residuals of linear estimation S&P 500) | Dimension | BDS statistics | P value |
| Epsilon=0.3 | 2 | 15.012 | 0.000 |
| | 3 | 23.858 | 0.000 |
| | 4 | 32.348 | 0.000 |
| | 5 | 42.397 | 0.000 |
| | 6 | 56.279 | 0.000 |

| Table 7 | Results for BDS test on the residuals of a non-linear estimation on 1 day return S&P 500 |
|-----------------|---------------------|---------------------|---------------------|
| BDS test (Residuals of non-linear estimation S&P 500) | Dimension | BDS statistics | P value |
| Epsilon=0.3 | 2 | 15.752 | 0.000 |
| | 3 | 24.364 | 0.000 |
| | 4 | 32.920 | 0.000 |
| | 5 | 43.289 | 0.000 |
| | 6 | 57.588 | 0.000 |
defining features for volatility from earlier literature. Subsequently, the preprocessing stage of each experiment, held on the data, selects the most contributing features.

4.2 Artificial Neural Network Hyperparameters

Deploying an efficient architecture of neural network is a tricky job. There are many new techniques to deal with problems in this topic, yet solving problems related to return prediction is hard because of the trap of local minimum. While dealing with real-world data scholars usually encounter the problem of overfitting, which is mainly due to getting trapped in local minimum or saddle points. Dealing with such problems, needs a good specification on the depth of the network, number of nodes in each layer, activation functions for each layer, efficient learning rate, related loss function, effective optimizer, etc.

In this study, despite the extreme simplifications in the architecture of the network, overfitting occurred at the early epochs, so we switched to $\text{tanh}^3$ activation function. Although tanh has the problem of saturation especially in deep architectures, it gains better loss and lower chance of overfitting than ReLU$^4$ in this problem. By using tanh, we further extend the complexity of the architecture by adding hidden layers and nodes in each layer. From the other side, excessively adding layers to the architecture caused the primary layer’s weights to saturate and stop improving. Furthermore, LSTM has recurrent activation function which is in charge of performing feature extraction from data. We use ReLU as the recurrent activation

---

3 Hyperbolic tangent.
4 Rectified Linear Unit.
function. Experiments shows that this combination of activation functions causes a better feature extraction, regularization and prevents early overfitting.

In the proposed architecture, Adam optimizer proves to be the best within gradient descent optimizers. This seems to be due to the benefits of both RMSprop and momentum in finding better direction and step size. We found batches of size 22 and decay rate of 0.007 suitable for learning. In order to find an efficient learning rate, we pre-trained the network with advantage of learning rate scheduler as a warmup training. In this pre-training, learning rate is changed by a defined callback in a specific range with a high resolution to find the rate with best smooth loss gradient.

The studied data is split into 3 sets of data. The first set is the training data with 5193 observations. The second set is the validation set with 488 observations which is used for stopping the network from further training. As the network is intentionally halted from training at a specific epoch in which the residuals of the validation series are minimized, there may be knowledge leakage occurred in the training. As a result, the trained network is further examined on completely unseen test data with 114 observations.

There are four main experiments conducted in this paper. As the target value is min max scaled to $-1$ and $1$, in addition to linear activation function in the output layer, tanh function is further experiment too. The model is optimized by the feedback of MAE loss function while backpropagating, but other metrics are also reported.

### 4.3 Experiments

#### 4.3.1 Experiment 1

In the first experiment, the generated explanatory variables are primarily fed to PCA in order to extract best series of variables which can maintain the maximum amount of explained variance within the primary series. PCA pre-extracted knowledge from the high dimensional data and reduced the dimensionality up to 17 series with 99% cumulative explained variance. Figure 7 shows the amount of explained variance by adding each extracted component. The 17 extracted series are then fed into a network with 4 hidden LSTM layers. The network is additionally utilized with dropout and batch normalization layers to prevent overfitting. Srivastava et al. (2014) used dropout procedure in 2014 and showed that the performance of neural networks significantly improves. In this experiment there are totally 92,250 parameters estimated.

#### 4.3.2 Experiment 2

The second experiment has RF supervised feature selection approach. As a result, the training set is primarily fed into a RF with 80 nodes. RF selected 109 series with the highest explanation impact on target value. Next, the chosen features are fed into a network with 4 hidden LSTM layers. This network is also utilized with dropout and batch normalization layers. There are totally 108,842 parameters estimated in this experiment.
4.3.3 Experiment 3

In the third experiment, LSTM network with a relatively higher number of parameters is deployed on plain generated features to see if features extracted by the network can better predict the return series. In this model, LSTM’s hidden layer’s main activation functions are all set to hyperbolic tangent, while recurrent activation functions of the blocks are set to ReLU. Recurrent activation functions play the role of saving enhancive extracted features while dismissing the disruptive features. This is automatically performed in the training process by giving higher and lower weights to the respective features. In this experiment there are 158,810 parameters estimated.

4.3.4 Experiment 4

The architecture of the last experiment is quite same as the third experiment. In this experiment features are denoised using wavelet transform before being fed to the neural network. To denoise the return signal of S&P500, wavelet of Daubechies family is used. In specific, DB3 is used to decompose the main signal into coefficient components. Next, the high-level component is passed through the non-negative garrote; see Gao (1998) threshold, and finally, the coefficients are used to reconstruct the signal. PyWavelets python package; see Lee et al. (2019) is used to apply various threshold values to the high-frequency coefficients. There is an optimization problem solved to maximize the noise extracted along with maintaining white noise characteristics. This is done in order to maximize the simplification of return series while preventing from losing data that has knowledge. This causes the features to lose some of their unnecessary volatilities. Figure 8 shows the distribution of the

![Figure 7](image)

**Fig. 7** Amount of cumulative explained variance within the initial generated features by adding each component (17 components explains 99.99% of the variance within the generated features) (color used)
extracted noise data from return series. Results from serial correlation test and BDS test significantly rejects the presence of knowledge in the extracted noise and result of Jarque–bera test also rejects absence of normality. As a result, the noise has the properties of a white noise and can be removed from the main return series. Figure 9 shows the return of S&P 500 before and after removing the extracted noise value.

Finally, features including denoised lags of the target value are fed into an LSTM network with 4 hidden layers. In this experiment totally 205,420 parameters are estimated.

### 4.3.5 Benchmark Method

Finally, as the basic and benchmark known method, an application with 4 layers of densely connected simple neural network is implemented. In this experiment there are no feature selection, feature extraction or even denoising approaches included. Instead, the raw generated explanatory variables that were fed into each of the 4 methods above are fed to this model. This is done in order to effectively compare each of the methods with this model.

### 5 Evaluation and Statistical Tests

To evaluate each experiment residuals, we use MAE, MSE, MAPE and R² score.

\[
\text{MAE} = \frac{\sum_{t=1}^{n} |A_t - F_t|}{n}. \tag{14}
\]

\[
\text{MSE} = \frac{1}{n} \sum_{t=1}^{n} (A_t - F_t)^2 \tag{15}
\]

\[
\text{MAPE} = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| \tag{16}
\]

Here \( A_t \) is the actual value and \( F_t \) is the forecast value. MAPE is an important metric because it is normalized to the scale of target value. This metric was proposed by Makridakis (1993) as it is more stable than other criteria. This important criterion can be used when researchers are trying to compare the accuracy of two different predictions in different time periods.

Although there are various conventional evaluation metrics (as mentioned above) which are generally used in prediction problems, they don’t adequately imply the profitability of financial forecasts. Therefore, evaluation metrics such as CSP\(^5\) and CDCP\(^6\) were introduced by Pesaran and Timmermann (1992). Raoofi et al. (2015), with regard to CSP and CDCP, ranked the ability of linear and non-linear models in predicting the sign and direction of returns on Tehran Stock Exchange Index. CSP in

---

\(^5\) Correct Sign Predictions.

\(^6\) Correct Direction Change Prediction.
The CDCP and CSP are calculated according to the following algorithms:

\[
CDFP = \frac{1}{T - (T_y - 1)} \sum_{t=T_i}^{T} z_{t+s}.
\]  

**IF** \((A_{t+s} - A_t)(F_{t,s} - A_t) > 0; \) **THEN**: \(z_{t+s} = 1;\)

**ELSE**: \(z_{t+s} = 0.\)

---

**Fig. 8** Extracted noise value from the return series of S&P 500–this noise lacks any knowledge and is meaningless, thus it can be removed from the data to avoid unnecessary complexity of estimations (color used)

**Fig. 9** Return series of S&P 500 vs denoised series of S&P 500–the denoised signal is feed to the model for prediction (color used)
\[
CSP = \frac{1}{T - (T_y - 1)} \sum_{t=T_1}^{T} z_{t+S}.
\] (18)

**IF** \((A_{t+S}) (F_{t,s}) > 0; \) **THEN:** \(z_{t+s} = 1,\)

**ELSE:** \(z_{t+s} = 0.\)

Here \(T\) is the total size of out of sample data and \(T_1\) is the first observation of out of sample data.

Furthermore, the Diebold-Mariano statistical test is applied to verify the accuracy precision difference on each pair of the models introduced in this paper. This test considers sequences of out-of-sample time series. The actual target value and two individual predictions are fed to the DM test, and the difference of predictions according to the target value is calculated. The null hypothesis is the condition at which the expected difference between the two predictions loss is statistically zero, keeping in mind that the differences in loss can be according to any given loss function. The DM statistics is calculated as follows:

\[
\text{DM} = \frac{\bar{d}}{\sqrt{\frac{2\pi\hat{f}_d(0)}{n}}},
\] (19)

\[
\bar{d} = \frac{1}{n} \sum_{t=1}^{n} (d_t)
\] (20)

Here \(d_t\) is the loss differential according to any function selected for the test, such as MSE or MAE, etc. And \(\hat{f}_d(0)\) stands for the spectral density of the loss differential at frequency 0.

Wilcoxon test, from the other side, is a non-parametric test used to test the significance of the differences of the median of two same size samples. As a result, the null hypothesis of this test is along with the null hypothesis of Diebold Mariano test. Rejecting the null hypothesis of this test shows that the two samples are significantly different by their median. The Wilcoxon signed rank test’s statistic is as follows:

\[
\text{WS} = \sum_{t=1}^{T} I_+(d_t) \text{rank}(|d_t|),
\] (21)

where \(I_+(d_t) = \begin{cases} 
1, & d_t > 0 \\
0, & \text{otherwise} 
\end{cases}\)

**6 Results and Discussion**

This research has aimed at step-by-step walking through market prediction. First, investigation on Efficient Market Hypothesis is performed by BDS test. This test rejected the presence of hypothesis indicating market efficiency in S&P 500. As a
result, price prediction model may be able to extract predictive knowledge from the training data. The test is applied on the return series of S&P 500 in the first step with 6-time lags. The epsilon term is selected as 0.3 for a precise investigation. After rejecting the EMH, the attempts for better predicting the market were held by 4 proposed preprocessing methods on LSTM.

Results from different preprocessing methods applied on deep neural network shows that feature extraction by PCA causes the best forecast on unseen data by MAE and CSP metrics. This seems to be due to the regularization of features extracted by PCA. Moreover, application of Random Forest and deep learning has minimized the loss by the results inferred from MAPE metric. From the other side, the application of deep learning with just its own feature extraction procedure has better forecasted direction changes in the return series by CDCP metric. Finally, application of wavelet and deep learning shows that it can’t minimize residuals as the other methods could.

Comparing the models with the benchmark method, only application of PCA and Deep learning has overcome it on unseen data regarding the MAE, MAPE and CSP score. This shows that not always complicated methods gain better accuracy, but they may also cause lower chance of success. Table 8 shows the detail of the prediction errors on validation and test set.

Figure 10 depicts the fit plot of Application of PCA and Deep Learning’s predictions and target value within test sequence.

In term of error diagnostics, the test sequence error of each method is defined in three main qualities. First is the Bias term of error, which is easier to deal with than standard deviation. Bias term of the error is the mean distance of predictions from target value. This term of error can be solved by incorporating this factor in the training process as a feature. Second is the standard deviation of the error. This criterion of the error value is not easy to solve. Application of Random Forest and deep learning has gained loss with lower standard deviation. Along with these qualities, in stock market predictions, the maximum loss is critical for investment analyzers as it implies the maximum draw down of a trading strategy. As a result, maximum loss is reported for each model. It seems that application of deep learning has achieved lowest maximum loss value during the test predictions. Table 9 reports the criteria of the implemented experiments which are important in error analysis of return prediction.

To statistically test the significance of the differences of the studied model’s accuracy, Diebold Mariano and Wilcoxon signed rank test is applied. Table number 10 shows the results of these tests. Results from Diebold Mariano test implies that none of the prediction models can significantly outperform the others in terms of MAE metric, but Wilcoxon signed rank test shows significant differences in the error of some of the models.

In summary, while verifying the proposed methods on test data, application of PCA and Deep Learning ranks first amongst the other methods. This is because of its better performance in terms of MAE, CSP score. Thus, this expert system can be the best choice in terms of predicting financial market.
7 Conclusion

In an overview, this paper has covered many aspects of the branch of financial market prediction. First of all, it has covered the evolutions of this branch since 1965 up to present. It contains the evolutions of the statistical approaches that tests the weak form of market efficiency and presence of information in historical data. Furthermore, the history of the development of Adaptive Market Hypothesis from Efficient Market Hypothesis and a modern comprehension from this notion is stated. Finally, the path from Adaptive Market Hypothesis to the recent market prediction approaches has been outlined. Besides, the importance of data characteristics analysis and Data Generation Process in financial predictions is emphasized with adequate statistical tests in this regard.

Secondly, the importance of feature engineering methods such as feature selection, feature extraction and denoising is further discussed. Hence, methods regarding various types of feature engineering are validated through four compiled models. Meanwhile, as the domination of LSTM neural networks in recent financial forecasting papers is significant, a densely connected neural network is implemented as the benchmark model. This is done to observe the performance of LSTM in comparison with DNN in extracting knowledge from financial time series.

Developed applications of Artificial Intelligence in finance has provided various services in academic and commercial fields, and AI has almost penetrated in every aspect of financial problems. Furthermore, with the quick developments in this field, the boundaries of expectations in the quality of provided solutions is further pushed back. From the other side, sources of financial data are quickly increasing. As a result, there should be a clear discrimination made between the ability of different preprocessing methods in extracting knowledge from definitive variables.

| Table 8  | Validation and test data results obtained from the compared experiments |
|----------|------------------------------------------------------------------------|
| Experiment                      | MAE       | MSE       | MAPE     | CSP      | CDCP     |
| Validation data                |           |           |          |          |          |
| PCA and deep learning          | 0.00990   | 0.00028   | 1.637    | 0.555    | 0.725    |
| RF and deep learning           | 0.00997   | 0.00028   | 1.312    | 0.567    | 0.721    |
| Deep learning                  | 0.00972   | 0.00028   | 1.429    | 0.581    | 0.737    |
| Wavelet and deep learning      | 0.00971   | 0.00025   | 1.502    | 0.575    | 0.725    |
| Benchmark method               | 0.00997   | 0.00028   | 1.406    | 0.567    | 0.721    |
| Test data                      |           |           |          |          |          |
| PCA and deep learning          | 0.00757   | 0.000106  | 1.169    | 0.552    | 0.719    |
| RF and deep learning           | 0.00783   | 0.0001042 | 1.089    | 0.543    | 0.719    |
| Deep learning                  | 0.00790   | 0.0001044 | 1.375    | 0.543    | 0.736    |
| Wavelet and deep learning      | 0.00810   | 0.000110  | 1.478    | 0.517    | 0.692    |
| Benchmark method               | 0.00783   | 0.000104  | 1.186    | 0.543    | 0.719    |
these reasons, the branch of market prediction is open and every year many papers are published in this topic.

In detail, the investigation conducted in this paper first tested the efficiency of S&P 500 market by various tests, including BDS test. After the tests rejected the IID behavior of return series, choosing appropriate target value for specific applications were discussed. Many literatures aim at volatility prediction for enhancing the quality of portfolio optimization, value at risk calculation, option pricing, etc. Some others may choose return values as the target value in order to support algo-trading bots for taking trading positions. This research aimed at predicting the 1-day return of S&P 500. Later, different expert applications were implemented to extract the knowledge that the earlier conducted statistical tests implied its presence. As mentioned in Table 10, results indicated that although the difference between prediction accuracy of the models is not significant with the implications of Diebold Mariano test, Wilcoxon signed rank test showed significant differences between the validated methods. Within the methods using LSTM neural network, PCA and LSTM neural network seem to have extracted more knowledge than the other competitors. But, the exaggerated performance of knowledge extraction of LSTM networks in comparison with dense neural networks was not inferred from the results. In another words, this study showed that although PCA and LSTM gained better results than Dense neural network, their difference was significant only with the Wilcoxon signed rank test.

![Fit plot of predicted values vs true values in application of principal component analysis and deep learning](color used)
| D-M test/Wilcoxon signed rank test | PCA and Deep Learning | RF and Deep Learning | Wavelet and Deep Learning | Deep Learning | Benchmark method |
|-----------------------------------|-----------------------|---------------------|--------------------------|---------------|------------------|
| PCA and Deep Learning             | \( P \text{ value}=0.000 \) | \( P \text{ value}=0.000 \) | \( P \text{ value}=0.908 \) | \( P \text{ value}=0.002 \) |
|                                   | \( \text{Wilcoxon stats}=1032.0 \) | \( \text{Wilcoxon stats}=1153.0 \) | \( \text{Wilcoxon stats}=3237.0 \) | \( \text{Wilcoxon stats}=2203.0 \) |
| RF and Deep Learning              | \( P \text{ value}=0.352 \) | \( P \text{ value}=0.543 \) | \( P \text{ value}=0.908 \) | \( P \text{ value}=0.002 \) |
|                                   | \( \text{D-M stats}=0.934 \) | \( \text{Wilcoxon stats}=3063.0 \) | \( \text{Wilcoxon stats}=468.0 \) | \( \text{Wilcoxon stats}=1958.0 \) |
| Wavelet and Deep Learning         | \( P \text{ value}=0.106 \) | \( P \text{ value}=0.214 \) | \( P \text{ value}=0.000 \) | \( P \text{ value}=0.000 \) |
|                                   | \( \text{D-M stats}=0.509 \) | \( \text{D-M stats}=1.247 \) | \( \text{Wilcoxon stats}=1292.0 \) | \( \text{Wilcoxon stats}=1634.0 \) |
| Deep Learning                     | \( P \text{ value}=0.236 \) | \( P \text{ value}=0.467 \) | \( P \text{ value}=0.402 \) | \( P \text{ value}=0.150 \) |
|                                   | \( \text{D-M stats}=1.191 \) | \( \text{D-M stats}=−0.840 \) | \( \text{Wilcoxon stats}=2769.0 \) | \( \text{Wilcoxon stats}=2769.0 \) |
| Benchmark Method                  | \( P \text{ value}=0.315 \) | \( P \text{ value}=0.943 \) | \( P \text{ value}=0.196 \) | \( P \text{ value}=0.405 \) |
|                                   | \( \text{D-M stats}=1.008 \) | \( \text{D-M stats}=−0.070 \) | \( \text{D-M stats}=−1.298 \) | \( \text{D-M stats}=−0.835 \) |

Those pair of models with significant variation of error term with 99 percent confidence interval are bold faced.
Furthermore, other three methods using LSTM could not even overperform the densely connected neural network. This research showed that there may be approaches to reduce prediction errors mainly by giving better representation of data to prediction algorithms like the application of PCA and LSTM, yet this field of study has a lot more uncovered areas left.

Furthermore, there must be a literature devoted to periods of market adaptation and prediction of market crisis period. It seems that after every uncertainty in market, adaptations occur in the system and EMH gradually develops. This research highly recommends the study of crisis analysis and duration of market uncertainty period with the methods that deep learning has provided. In addition, this study recommends including other data sources to the variables used in this research such as microblogging websites posts related to financial market for improving prediction accuracy.

**Funding** The authors did not receive support from any organization for the submitted work.

**Data Availability** The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

**Code availability** The codes related to the current study are available from the corresponding author on reasonable request.

**Declarations**

**Conflict of 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.

**References**

Abeysekera, S. P. (2001). Efficient markets hypothesis and the emerging capital market in Sri Lanka: Evidence from the Colombo Stock Exchange – A note. *Journal of Business Finance & Accounting, 28*(1–2), 249–261. https://doi.org/10.1111/1468-5957.00373

Adya, M., & Collopy, F. (1998). How effective are neural networks at forecasting and prediction? A review and evaluation. *Journal of Forecasting, 17*(5–6), 481–495.

Aminimehr, A., Raoofi, A., Aminimehr, A., & Aminimehr, A. (2021). The role of feature engineering in prediction of tehran stock exchange index based on LSTM. *Iranian Journal of Economic Studies, 9* (2), 527–548. https://doi.org/10.22099/ijes.2021.39877.1739

Baillie, R. T., Bollerslev, T., & Mikkelsen, H. O. (1996). Fractionally integrated generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics, 74*(1), 3–30. https://doi.org/10.1016/S0304-4076(95)01749-6

Barber, B. M., & Odean, T. (2000). Trading is hazardous to your wealth: The common stock investment performance of individual investors. *The Journal of Finance, 55*(2), 773–806. https://doi.org/10.1111/0022-1082.00226

Bari, O., & Agah, A. (2018). Ensembles of text and time-series models for automatic generation of financial trading signals from social media content. *Journal of Intelligent Systems. https://doi.org/10.1515/jisys-2017-0567*

Basu, S. (1977). Investment performance of common stocks in relation to their price-earnings ratios: A test of the efficient market hypothesis. *The Journal of Finance, 32*(3), 663–682. https://doi.org/10.2307/2326304

Bildirici, M., & Ersin, Ö. Ö. (2009). Improving forecasts of GARCH family models with the artificial neural networks: An application to the daily returns in Istanbul Stock Exchange. *Expert Systems with Applications, 36*(4), 7355–7362. https://doi.org/10.1016/j.eswa.2008.09.051
Huang, J., Chai, J., & Cho, S. (2020). Deep learning in finance and banking: A literature review and classification. *Frontiers of Business Research in China, 14*(1), 13. https://doi.org/10.1186/s11782-020-00082-6

Hyup Roh, T. (2007). Forecasting the volatility of stock price index. *Expert Systems with Applications, 33*(4), 916–922. https://doi.org/10.1016/j.eswa.2006.08.001

Jensen, M. C. (1978). Some anomalous evidence regarding market efficiency. *Journal of Financial Economics, 6*(2), 95–101. https://doi.org/10.1016/0304-405X(78)90025-9

Jin, Z., Yang, Y., & Liu, Y. (2020). Stock closing price prediction based on sentiment analysis and LSTM. *Neural Computing and Applications, 32*(13), 9713–9729. https://doi.org/10.1007/s00521-019-04504-2

Jurgovsky, J., Granitzer, M., Ziegler, K., Calabretto, S., Portier, P.-E., He-Guelton, L., & Caelen, O. (2018). Sequence classification for credit-card fraud detection. *Expert Systems with Applications, 100*, 234–245. https://doi.org/10.1016/j.eswa.2018.01.037

Kanas, A., & Yannopoulos, A. (2001). Comparing linear and nonlinear forecasts for stock returns. *International Review of Economics & Finance, 10*(4), 383–398.

Keerthi Vasan, K., & Surendiran, B. (2016). Dimensionality reduction using principal component analysis for network intrusion detection. *Perspectives in Science, 8*, 510–512. https://doi.org/10.1016/j.pisc.2016.05.010

Kim, H. Y., & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications, 103*, 25–37. https://doi.org/10.1016/j.eswa.2018.03.002

Kim, K.-J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*(1), 307–319. https://doi.org/10.1016/S0925-2312(03)00372-2

Kim, K.-J. (2006). Artificial neural networks with evolutionary instance selection for financial forecasting. *Expert Systems with Applications, 30*(3), 519–526. https://doi.org/10.1016/j.eswa.2005.10.007

Ko, P.-C., & Lin, P.-C. (2006). An evolution-based approach with modularized evaluations to forecast financial distress. *Knowledge-Based Systems, 19*(1), 84–91. https://doi.org/10.1016/j.knosys.2005.11.006

Ko, P.-C., & Lin, P.-C. (2008). Resource allocation neural network in portfolio selection. *Expert Systems with Applications, 35*(1), 330–337. https://doi.org/10.1016/j.eswa.2007.07.031

Kristjanpoller, W., & Minutolo, M. (2015). Gold price volatility: A forecasting approach using the Artificial Neural Network–GARCH model. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2015.04.058

Kvamme, H., Sellereite, N., Aas, K., & Sjursen, S. (2018). Predicting mortgage default using convolutional neural networks. *Expert Systems with Applications, 102*, 207–217. https://doi.org/10.1016/j.eswa.2018.02.029

LeCun, Y., & Bengio, Y. (1998). Convolutional networks for images, speech, and time series. In *The handbook of brain theory and neural networks* (pp. 255–258). MIT Press.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lee, G., Gommers, R., Waselewski, F., Wohlfahrt, K., & O’Leary, A. (2019). PyWavelets: A Python package for wavelet analysis. *Journal of Open Source Software, 4*(36), 1237. https://doi.org/10.21105/joss.01237

Li, B., & Sim, K. C. (2013). Improving robustness of deep neural networks via spectral masking for automatic speech recognition. 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 8–12.

Li, X., & Tang, P. (2020). Stock index prediction based on wavelet transform and FCD-MLGRU. *Journal of Forecasting, 39*(8), 1229–1237. https://doi.org/10.1002/jfor.2682

Lo, A. W. (1991). Long-term memory in stock market prices. *Econometrica, 59*(5), 1279–1313. https://doi.org/10.1017/2938368

Lo, A. W. (2004). The adaptive markets hypothesis. *The Journal of Portfolio Management, 30*(5), 15. https://doi.org/10.3905/jpm.2004.442611

Lu, C.-J. (2010). Integrating independent component analysis-based denoising scheme with neural network for stock price prediction. *Expert Systems with Applications, 37*, 7056–7064. https://doi.org/10.1016/j.eswa.2010.03.012
Makridakis, S. (1993). Accuracy measures: Theoretical and practical concerns. *International Journal of Forecasting, 9*(4), 527–529. https://doi.org/10.1016/0169-2070(93)90079-3

Mandelbrot, B. B. (1997). The variation of certain speculative prices. In B. B. Mandelbrot (Ed.), *Fractals and Scaling in Finance* (pp. 371–418). New York, NY: Springer. https://doi.org/10.1007/978-1-4757-2763-0_14

McAleer, M., & Medeiros, M. C. (2011). Forecasting realized volatility with linear and nonlinear univariate models. *Journal of Economic Surveys, 25*(1), 6–18. https://doi.org/10.1111/j.1467-6419.2010.00640.x

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics, 5*(4), 115–133. https://doi.org/10.1007/BF02478259

Murali, Y., Babu, M., Subramanyam, D., & Prasad, D. (2012). PCA based image denoising. *Signal & Image Processing. https://doi.org/10.5121/sipij.2012.3218

Nelson, D. B. (1991). Conditional heteroskedasticity in asset returns: A new approach. *Econometrica, 59*(2), 347–370. https://doi.org/10.2307/2938260

Ozbagyolgu, A. M., Gudelek, M. U., & Sezer, O. B. (2020). Deep learning for financial applications: A survey. *Applied Soft Computing, 93*, 106384. https://doi.org/10.1016/j.asoc.2020.106384

Park, J.-I., Lee, D.-J., Song, C.-K., & Chun, M.-G. (2010). TAIEX and KOSPI 200 forecasting based on two-factors high-order fuzzy time series and particle swarm optimization. *Expert Systems with Applications, 37*(2), 959–967. https://doi.org/10.1016/j.eswa.2009.05.081

Pele, D. T., & Voinaegu, V. (2008). Testing market efficiency via decomposition of stock return. Application to Romanian Capital Market. *Journal for Economic Forecasting, 5*, 79–136.

Pesaran, H., & Timmermann, A. (1992). A simple nonparametric test of predictive performance. *Journal of Business & Economic Statistics, 10*, 561–565. https://doi.org/10.1080/07350015.1992.10509922

Qi, M. (1996). 18 Financial applications of Artificial Neural Networks. *Statistical Methods in Finance* (pp. 529–552). Elsevier. https://doi.org/10.1016/S0169-7161(96)14020-7

Quandt, R. E. (1972). A new approach to estimating switching regressions. *The American Statistical Association, 67*(3), 306–310. https://doi.org/10.1080/01621459.1972.10482378

Raoofi, A., & Mohammadi, T. (2018). Forecasting tehran stock exchange index returns using a combination of wavelet decomposition and adaptive neural fuzzy inference systems. *Iranian Economic Research, 23*(76), 107–136.

Raoofi, A., Zarra-Nezhad, M., & Bayani, O. (2015). Assessment and Comparison of linear and non-linear Methods for Forecasting Returns on Stock Market Index. *International Journal of Scientific Management and Development, 3*, 1–9.

Rather, A. M., Agarwal, A., & Sastry, V. N. (2015). Recurrent neural network and a hybrid model for prediction of stock returns. *Expert Systems with Applications, 42*(6), 3234–3241. https://doi.org/10.1016/j.eswa.2014.12.003

Rosenberg, B., Reid, K., & Lanstein, R. (1985). Persuasive evidence of market inefficiency. *The Journal of Portfolio Management, 1*(1), 9. https://doi.org/10.3905/jpm.1985.409007

Samuelson, P. A. (2015). Proof that Properly Anticipated Prices Fluctuate Randomly. In A. G. Malliaris & W. T. Ziemba (Eds.), *The World Scientific Handbook of Futures Markets* (pp. 25–38). World Scientific. https://doi.org/10.1142/9789814566926_0002

Sang, C., & Di Pierro, M. (2019). Improving trading technical analysis with TensorFlow Long Short-Term Memory (LSTM) Neural Network. *The Journal of Finance and Data Science, 5*(1), 1–11. https://doi.org/10.1016/j.jfds.2018.10.003

Sharif, A., Aloui, C., & Yarovaya, L. (2020). COVID-19 pandemic, oil prices, stock market, geopolitical risk and policy uncertainty nexus in the US economy: Fresh evidence from the wavelet-based approach. *International Review of Financial Analysis, 70*, 101496. https://doi.org/10.1016/j.irfa.2020.101496

Shiller, R. J. (1981). Do stock prices move too much to be justified by subsequent changes in dividends? *The American Economic Review, 71*(3), 421–436.

Shleifer, A. (2000). *Inefficient Markets: An Introduction to Behavioural Finance*. UK: Oxford University Press.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(1), 1929–1958.
Ta, V.-D., Liu, C.-M., & Tadesse, D. A. (2020). Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *Applied Sciences, 10*(2), 437. https://doi.org/10.3390/app10020437

Tin Kam, H. (1995). Random decision forests. Proceedings of 3rd International Conference on Document Analysis and Recognition, pp 14–16.

Tsaih, R., Hsu, Y., & Lai, C. C. (1998). Forecasting S&P 500 stock index futures with a hybrid AI system. *Decision Support Systems, 23*(2), 161–174. https://doi.org/10.1016/S0168-9236(98)00028-1

Tyrulis, H., & Papacharalampous, G. (2017). Variable selection in time series forecasting using random forests. *Algorithms, 10*(4), 114. https://doi.org/10.3390/a10040114

Vukovic, D., Vyklyuk, Y., Matsiuk, N., & Maity, M. (2020). Neural network forecasting in prediction Sharpe ratio: Evidence from EU debt market. *Physica A: Statistical Mechanics and its Applications, 542*, 123331. https://doi.org/10.1016/j.physa.2019.123331

Wang, W., Li, W., Zhang, N., & Liu, K. (2020). Portfolio formation with preselection using deep learning from long-term financial data. *Expert Systems with Applications, 143*, 113042. https://doi.org/10.1016/j.eswa.2019.113042

Wang, Y.-H. (2009). Nonlinear neural network forecasting model for stock index option price: Hybrid GJR-GARCH approach. *Expert Systems with Applications, 36*, 564–570. https://doi.org/10.1016/j.eswa.2007.09.056

Werner Kristjanpoller, R., & Esteban Hernández, P. (2017). Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors. *Expert Systems with Applications, 84*, 290–300. https://doi.org/10.1016/j.eswa.2017.05.024

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems, 2*(1), 37–52. https://doi.org/10.1016/0169-7439(87)80084-9

Yao, Y., Zhai, J., Cao, Y., Ding, X., Liu, J., & Luo, Y. (2017). Data analytics enhanced component volatility model. *Expert Syst. Appl., 84*(C), 232–241. https://doi.org/10.1016/j.eswa.2017.05.025

Yavuz, M., Sakarya, Ş., Ozdemir, N., & Karaoğlan, A. (2015). Stock market index prediction with neural network during financial crises: a review on bist-100. *Financial Risk and Management Reviews, 1*, 53–67. https://doi.org/10.18488/journal.89/2015.1.2/89.2.53.67

Zhong, X., & Enke, D. (2019). Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financial Innovation, 5*(1), 24. https://doi.org/10.1186/s40854-019-0138-0

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Authors and Affiliations**

Amin Aminimehr¹ · Ali Raoofi² · Akbar Aminimehr³ · Amirhossein Aminimehr⁴

Amin Aminimehr
aminaminimehr@outlook.com

Ali Raoofi
aliraooif1@gmail.com

Akbar Aminimehr
aminimehr_a@yahoo.com

Amirhossein Aminimehr
amir_aminimehr@comp.iust.ac.ir

¹ Department of Management, Ershad Damavand Institute of Higher Education, Vesal Shirazi St, Enghelab St, No 28, 26Thstreetstreet, Kuy e Nasr, Tehran 14168-34311, Iran
2 Allameh Tabataba’i University Faculty of Economics, Economics College of Allameh Tabatabae’i University, Corner of Ahmad Qasir St., Beheshti St., Tehran 15136-1541, Iran

3 Accounting, Management and Economic Department, Payame Noor University, Nakhl St, Lashkarak Highway, Tehran 14556-43183, Iran

4 School of Computer Engineering, Iran University of Science and Technology, University St, Hengam St, Resalat Square, Tehran 13114-16846, Iran