Journal of Applied Data Sciences
Vol. 2, No. 4, December 2021, pp. 109-134

Implementation of ANN and GARCH for Stock Price Forecasting

Hendra Mayatopani [1,*]

[1] Information Systems Program, Pradita University, Indonesia
[1]hendra.mayatopani@pradita.ac.id*;
* corresponding author

(Received: August 4, 2021; Revised: September 14, 2021; Accepted: October 5, 2021; Available online: December 31, 2021)

Abstract

For simulating intricate goalfunctions, neural networks are a technology that is employed in artificial intelligence. The usage of artificial neural networks is becoming more popular.(ANNs) to certain sorts of tasks, for example learning to comprehend complicated sensor data collected in the real world, is one of the most effective methods of learning approaches available. The usage of time series models in financial time series prediction has grown significantly over the past decade, and their relevance in this area continues to expand. To be more specific, the goal of this research is to determine whether neural networks have the ability to predict financial time series in general, or, more specifically, whether they have the ability to predict future patterns i The stock market in the United States is characterized by the European Union, and Brazil, among other things. They are compared to a well-known forecasting approach, generalized autoregressive conditional heteroskedasticity, in this research, and their accuracy is shown to be superior (GARCH). Aside from that, the optimal network design for each data sample is developed for each data sample. According to this article, ANNs are capable of forecasting the stock markets under examination, and their resilience may be increased by varying the network topology utilized to construct them. Aside from that, the results of this research demonstrate that ANNs outperform GARCH models in terms of efficiency of statistical performance.

Keywords: ANN; GARCH; Stock Price; Data Mining.

1. Introduction

Financial and economic modeling have a long history of study. This methodology makes use of time series analysis, which has shown to be one the most durable and extensively utilized techniques available in the area. When it comes to understanding the behavior of time series, there are two types of models to choose from. Linear models are the first kind of model that is introduced. For the most part, in time series analysis, a linear technique is represented in one of the following ways: A few examples of statistical approaches are Box Jenkins procedures, Kalman filters, Brown's exponential smoothing theory, and piecewise regression, amongst others. Another kind of model is nonlinear, which includes Takens theorem-based models, Markov switching models, threshold autoregression models, and smooth transition models are all examples of stochastic switching models. autoregressive models, to mention a few examples. Nonlinear models include the following: These algorithms make an attempt to predict future values by reconstructing the time series from a sample of the data that is now accessible to the algorithm. The fact that these tactics are statistically sound does not make up for the reality that they have a poor track record of performance when applied to the forecast of financial markets.

Although recent data indicates that financial markets are nonlinear, the linear approaches stated above are still capable of adequately describing the nonlinear systems seen in financial market time series analysis [1]. Huang et al., [2] offer a good overview of the occurrence of nonlinearities in financial data and construct a model for forecasting financial time series termed generalized autoregressive conditional heteroskedasticity (GARCH), which incorporates all of the observable properties in these series. Although the economy evolves (though gradually) over time, one element that is not possible adequately represented by linear models with a fixed specification emerges: the rise as well as the decline in the stock market. the economy changes with time as a developing coefficient estimate. Numerous variables have a role in finance and economics,political developments, general economic conditions, and

so on, as well as trader expectations. As a consequence, financial projections are more accurate. and economics is characterized by motions exceedingly challenging.

In current quantitative finance, artificial neural networks (ANNs) have developed as a powerful statistical modeling technique. They provide an attractive alternative tool for both academics and practitioners. They can recognize patterns, classify data, evaluate it, model it, forecast it, and regulate it [1-4]. Numerous differentiating characteristics of ANNs contribute to their value and attractiveness in predicting. To begin, ANNs are nonlinear data-driven techniques. They are capable of modeling nonlinear systems without being aware of the relationships between the input and output variables. The nonparametric ANN model may be favored over typical parametric statistical models in cases when the input data do not match the parametric model's assumptions or when the dataset contains big outliers [5]. Second, ANNs are approximations of universal functions. A neural network has been proven to be capable of approximating any continuous function to any required degree of accuracy [6]. Thirdly, ANNs exhibit generalizability. ANNs are often used once they have learned from the data that has been supplied capable of accurately inferring the unknown a subset of a population, even if the sample data contains erroneous information It is possible to use neural networks to collect time series data while keeping the underlying pattern or autocorrelation structure intact, even when the fundamental rule guiding the system is unknown or unarticulated.

Due to their pattern identification capabilities, ANNs have been effectively employed in a wide variety of sectors and are rapidly being used in economics and business research. Ostermark [7] classify relevant articles according to their publication year, application area, journal, various decision characteristics (problem domain, decision process phase, management level, task interdependence), development method, integration with other technologies, and significant contribution. Rathnayaka et al., [8] review works that discuss modeling challenges associated with the use of ANNs to forecasting. The authors list the ANN models' most often noted benefits and downsides. Liau and Reid [9] discuss the ANN system and its widespread use in financial markets. Additionally,Their research stresses the benefits of artificial neural networks (ANNs) over more traditional techniques. Following that, the study proceeds with an examination of numerous financial firms' effective usage of ANNs. Seetharam and Reid [10] discuss the overall methods necessary to create, train, and test a neural network using commercially available software in their book Neural Networks for Financial Forecasting. Additionally, Shapiro [11] discusses how neural networks, fuzzy logic, and evolutionary algorithms are used in financial markets.

Lie and Tan [3], Zhou et al. [11], and Ostermark [7] all use ANNs to forecast the stock market. Loo [12] demonstrate that their ANN model with monetary fundamentals cannot outperform the random walk model. Alternatively, Jarrett [13] uses ANN models to analyze macroeconomic time series and finds that they outperform both linear and other nonlinear models. Jarrett [14] investigated the efficacy of utilizing ANNs in conjunction with external data like commodities prices and currency exchange rates in forecasting Dow Jones Industrial Average index fluctuations. According to their findings, there are a number of advantages to adopting these indicators rather than traditional approaches or more conventional strategies that are focused only on historical data creation and analysis.

A study conducted using ANNs to replicate the Indian stock market was conducted by Rathnayaka et al. [8]. (price index). When it comes to replicating the close of the Bombay Stock Exchange's Sensex index, the closing values of the authors investigate the usage of artificial neural networks (ANNs). Utilizing two different types of artificial neural networks, the researchers compare their performances. RMSE and MAE are two forms of errors that might occur. The root mean squared error (RMSE) is the most common type of mistake (ANNs). According to their findings, they found that increasing the input values to the ANN may improve the quality of the verified outcomes. A recent prediction study of the Ibovespa conducted by Liau and Reid [9] made use of ANNs and an adaptive exponential smoothing approach. Among the objectives of the research were to compare the forecasting skills of the two systems, and to determine how accurate they were when it comes to forecasting the direction returns on the stock market in the short term. Both strategies provide findings that are equivalent when it comes to anticipating index returns, as the authors show in their paper. In order to test forecasting accuracy, the researchers utilize two metrics: root mean square error (RMSE) and N(tend), which quantify the model's ability to acquire suitable tendencies, and/or the

number of times the predictions match the market's genuine tendencies, respectively, Finally, Ostermark [15] contrasted the profitability of applying ANN predictions translated into a fundamental trading strategy to the profitability of employing a normal buy–hold strategy in his research. In the end, he discovered that the first option was more lucrative. The authors apply this method to the The Taiwan Weighted Index (TWI) and the Standard & Poor's (S&P) 500 were studied, and it was shown that trading rules based on artificial neural networks (ANNs) outperformed the buy–hold strategy in both indexes.

However, these studies have concentrated on It is possible to trade markets that correspond to particular kinds of ANN architectures without necessarily comparing them to more classic statistical approaches like the autoregressive integrated moving average (ARIMA)–GARCH models, which are used to forecast the S&P 500 and the Nikkei 225 Index, respectively. The fact that no statistical comparisons are made across competing approaches is another disadvantage of these research, which instead depend on existing performance measurements For example, the root mean square error (RMSE) and the mean absolute error (MAE) are used to determine effectiveness (MAE)

To be more specific, the purpose of The purpose of this paper is to analyze and investigate the use of neural networks to forecast future trends in North American stock market indexes, specifically the Dow Jones and S&P 500 in the United States, the DAX in Germany, the CAC 40 in France, the FTSE in Britain, the IBEX 35 in Spain, the PSI 20 in Portugal, and the TSX in Canada and the United Kingdom the Dow Jones and S&P 500 in Germany, as well as the Dow Jones and S&P 500 in Germany, among others (Brazil). A thorough examination analysis On the use of neural networks in various applications in the NASDAQ (stock market) to anticipate Indicators of the economy is provided by the authors in this research. Artificial neural networks (ANNs) trained on a GARCH model and applied to each series in this research are evaluated using standard performance metrics AGS and MGN tests, for example, are statistical tests that are used to determine whether or not a person is a serial killer. In this paper, we want to demonstrate the robustness of the model by looking at how artificial neural networks (ANNs) may be used to account for heteroskedasticity in time series in the financial sector.

2. Literature Review

2.1. Applications in Stock Market Index Forecasting

A large expected return on investment makes the stock market one of the most attractive venues to invest money because of the high projected return on investment. The expected payoff, on the other hand, is proportionate to the amount of risk that has been specified. According to a number of studies, the stock market may be a tough environment in which to execute one's commercial operations. It may be divided into three categories based on the degree to which the market functions efficiently. The efficient market theory, in its most extreme version, states that, in the absence of fresh information, the market price of a security rapidly integrates all previously available information. As a result, all price forecasters, regardless of whether or not they have access to private data, are wasting their time and resources. According to the semi-strong variety of the efficient market hypothesis, on the other hand, owners of private knowledge maintain their ability to benefit from the information they have acquired over time. In its most basic form, this statement says that all information gained via the study of prior trade history is reflected in the price of the product under consideration. Without a doubt, historical trade data is available, which indicates that the The efficient market in its weakest form hypothesis This subset of the semi-strong version of the formula hypothesis, which is a subset of the strong form of the hypothesis, The strong form, in turn, is a subset of the weak form of the hypothesis.

A complicated series of events results in fluctuations in the stock market, the effects of which are represented as a mixture of profits and losses in a stock market time series, which may then be forecast using extrapolation techniques. Every year, we see periodic swings in the economy that are produced by seasonal patterns and economic cycles. These patterns and cycles are responsible for the periodic swings in the economy that we see every year. Stock market fluctuations over a short period of time and from from day to day occur They occur at random and are

difficult to forecast; however, because They are a group of people unpredictable and unpredictable, they frequently account for the majority of profits and losses in stock trading, particularly for day traders, because they are the most volatile and unpredictable.

As a consequence of multiple study investigations, a number of The use of decision assistance systems has been shown developed with the goal of giving investors with optional projections for their investments as a result of the findings of these studies. Historically, market gurus have used technical analysis to increase the accuracy of their stock market forecasts. They have done so for a number of reasons, including improving the accuracy of their predictions. Technical analysis, in general, is a technique for identifying stock movements that involves evaluating the stock's historical value over a period of time. Historically based support and resistance levels may be established directly, or history can be incorporated into a number of technical indicators to offer further insight into the price movement. Time series analytic techniques as an example, mixed moving average with an autoregressive component (ARMA) and multiple regression models,which are based on time series data have been shown to be typically put to use to tackle this problem in the literature. Significant evidence suggests that the price of stocks, at least in the near term, can be anticipated to a certain extent, at least in theory.

2.2. Input Variables

The identification of the input variables for neural network-based stock market index forecasting may be accomplished using one of two theoretical approaches: The first chapter investigates the relationship between the price of the stock market index and other macroeconomic indicators. The second alters the relationship between stock prices, dividends, and trading volume by introducing nonlinearity into the equation.

Ostermark [16] is a researcher who studies the link between changes in financial investment opportunities and changes in the economy. In addition, in the author's own words demonstrates that the default spread, term spread, one-month spread, and so on Treasury rate of bill-paying, delayed industrial output The growth rate and the dividend–price ratio are both important all significant predictors with respect to a future stock market index according to the author. It is argued in this research that these indicators have the potential to be utilized to anticipate future stock market indices because of their association to changes in the macroeconomic environment. Yin et al. [17] identify three main risk variables that appear to explain average stock and bond returns: the general market component, firm-size-related factors, and book-to-market equity. The general market component is defined as the component of the overall market that includes all stocks and bonds. When elements such as the lagged stock index and past interest rates are not included, as shown by Shou et al. [11], wrong results might be drawn. Mohanty and colleagues [18] investigate the predicting ability of time delay neural networks for the S&P 500 index time series.

A method known as vector autoregression (VAR) is the most often used when attempting to understand the associations between two variables. Because it allows for the simultaneous investigation of several variables as well as the automated evaluation of their relationship using advanced statistically significant criteria, it has the advantage of being time-efficient. In accordance with Ao (2003a and b), the following are the outcomes: (1) Historical prices, JP, the NASDAQ, the S&P, and the DJ are all important; (2) Historical prices, the NASDAQ, the S&P, and the DJ are also important; (3) Historical prices, JP, the HK, the NASDAQ, the S&P, and the DJ are all important; (4) Historical prices, JP, the NASDAQ, the S&P, and the DJ are all important; (5) DJ is important; (6) DJ is important; (7) DJ is important; It was revealed via the use of VAR modeling that the the stock's historical worth on its own in addition to the stock's movement in the United States, are the most significant pieces of information for Asian markets, according to the findings. If you are attempting to construct a correlation diagram between the local and national markets, it is also helpful to be aware of the components of market dynamics that are affected by volume and time variation. According to the available statistics, the Hong Kong market (and, to a lesser degree, a variety of other Asian markets) is driven by use of local concerns for example, currency issues rather than global concerns during

periods of low correlation, such as the Asian financial crisis of the late 1990s. At times, the local market may be similar to those found in the United States, while at other times, it may be completely different from those found in the United States.

If we're talking about macroeconomic indicators, we're talking about the The term structure of interest rates (TS-IR), the short-term interest rate (ST-IR), and the long-term interest rate are all terms that are used to describe interest rates. Consumer price index (CPI), industrial production (IP), government consumption (GC), private consumption (PC), gross national product (GNP), and gross domestic product (GDP) are all measures of economic activity (GDP) are among the macroeconomic indicators that are taken into consideration (GDP). Input factors such as these are the most evident ones that a forecaster is likely to come across throughout the course of his or her inquiry. In spite of the fact that additional macroeconomic indicators may be used, the majority of researchers believe that Interest rates and lagged predictive factors are two examples of lagged predictive variables that give the great majority of the crucial information for projecting future inflation rates. According to some study, the distinction between long and short-term and rates on short-term bonds, which is defined as the difference between long-term and Bond yields, particularly short-term bond rates, may have significant predictive value in predicting the performance of a stock index.

The historical data for each of the series under consideration, which are being used as input variables in this study, is shown in Table 1. This study's main goal is to discover whether or not the ANN structure has an effect on the predicted results. Following the identification of the optimal We evaluate the structure by comparing it to outputs of the ANN and GARCH, respectively models in order to evaluate which structure is the most effective.

Table. 1. The stock market indicators that make up the sample are listed below.

Indicator	Country	Variable
Dow Jones	US (United States)	DOW
FTSE	UK (United Kingston)	FTSE
IBEX 35	SP (Spain)	IBEX
PSI 20	PR (Portugal)	PSI
Ibovespa	BR (Brazil)	IBOV
S&P 500	US (United States)	S&P
DAX	GR (Germany)	DAX
CAC 40	FR (France)	CAC

2.3. Models of the GARCH

steps: An important disadvantage of ARIMA models is that they are based on the assumption that the modeled variable's volatility4 (for example, stock price) stays constant over time. This assumption is incorrect. It's unfortunate, but this is not always the case. Considerable variances in one direction are sometimes accompanied by significant discrepancies in the other direction as well. To put it another way, it seems as if the volatility of asset return returns is serially connected with each subsequent return. This portion of the financial time series could only

be captured by using the autoregressive conditional heteroskedasticity (ARCH) model, which had to be constructed. The ARCH5 procedure is divided into the following steps:

$$ARCH\ (q)\colon\ y_t = \alpha + \sigma_t \varepsilon_t \quad (1)$$

$$\sigma_t = \sqrt{\alpha_0 + \sum_{i=1}^{q} \alpha_i y_{t-i}^2} \quad (2)$$

σ_t denotes the conditional standard deviation of y_t , given the process's previous values, and a denotes a dependable characteristic Because the ARCH(q) process is a non-correlated process with a mean and variance that are constant, (α_0)Variance that is continuous and unconditional, but a non-constant conditional variance.

It is simple to interpret this strategy as a model for volatility clustering, which is what we will do here. The presence of large values of ($y^2_{t=i}$) historical squared returns results in a significant increase in current volatility. The ARCH(q) model is a subset of the GARCH(p,q) model, which may be found here and consists of the following elements:

$$GARCH\ (p, q)\colon\ y_t = a + \sigma_t \varepsilon_t \quad (3)$$

$$\sigma_t = \sqrt{\alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2} \quad (4)$$

As part of the process of developing this model, the present volatility is determined by averaging the volatilities over the previous q a number of days as well as the squared returns over the previous p days. Following in the footsteps of the pioneers Following the work of Engle and Bollerslev are two of the most well-known names in the world of sports (the creators of the ARCH and GARCH models, respectively), a brief but active field of research has developed, resulting in a variety of Variations of the GARCH(p,q) model, including power GARCH (PGARCH) models, exponential GARCH (EGARCH) models, threshold GARCH (TGARCH) models, and others that integrate so-called leverage effects, are available (e.g., power GARCH models), among others It is possible to generate a more exact simulation of the well-documented asymmetric behavior of return, in which a price increase in reaction to a product is seen, via the application of leveraging terms in financial models. On the other hand, a price decline in reaction to unfavorable news leads to an increase in the volatility of the market in the succeeding period. It is recommended that additional GARCH models be included since they capture unique short- and long-term volatile patterns, as well as GARCH-in-the-mean models (GARCH-M), which allow for the estimation of for the calculation of volatility-dependent mean values of return.

2.4. Neural Networks

Approximating target functions utilizing real-, discrete-, and vector-valued inputs is made possible by a neural network is a kind of network that consists of neurons and connects them together based on methods, which provide a rigorous approach for doing so. When it comes to particular types of tasks, such as learning to grasp complex real-world sensor data, artificial neural networks (ANNs) are among the most effective learning approaches available. For the most part, artificial neural networks (ANNs) were developed to facilitate this Distributed computing is used in this kind of parallel processing calculation representation. However, although the majority of ANN software is designed to operate on serial computers that mimic distributed processes, more efficient implementations of the algorithms have been built on highly parallel and specialized systems.

2.4.1. Definitions at a Basic Level

The multilayer perceptron structure is the kind of structure that is most often seen in artificial network topologies (MLP). There are a lot of processing units layered on top of each other, and they are all linked (also termed neurons or nodes). As previously stated, a collection of values (input data) is fed into a series ofNeurons in the so-called input layer, also known as the prefrontal cortex, where they are processed before being delivered via a sequence of neurons in the hidden layer, where the processing is completed. Weight is a feature that differentiates each connection and is responsible for defining the total strength of the connection. Weight is measured in grams (or weakness). An artificial neural network may be trained to convert patterns when it comes to the input layer into acceptable values in the output layer under certain situations, which can be accomplished by adjusting the weights in a specific way. It is possible to do weight adaptation via the use of a learning or training algorithm, which will be discussed in further depth later in this text.

Once the training data has been partitioned, it is generally separated into (at least) two non-overlapping parts, which are referred to as the training and testing sets, respectively. It is usual practice to teach the network the intended target function via the use of a substantial training set, which is then used to Create a network that can be trained to perform the required goal function. It is used to determine generalization ability, which is defined as the network's ability to make reliable inferences about the data's population characteristics from the sample characteristics of the training set, after it has been trained on the training set's data. The network is trained on the training set's data and then applied to the test set's data to determine generalization ability. After the network has been trained on the data from the training set, It is applied to the data from the test set in order to determine whether or not it has generalization potential (e.g., if a network has to learn a sine function, it should produce correct results for all real numbers and not only for those in the training set). Having overfitted means that a network has been unable of learning to generalize and has instead learned just the specific properties of the training patterns while failing to detect or recognize the broad elements of the input (i.e., the network produces correct results for training patterns while producing a high error rate in the test set).

2.4.2. The Characteristics of a Neural Network

Particularly well-suited to situations in which noisy, complicated sensor data is needed, such as that provided by cameras and microphones, among other sources. When it comes to decision tree learning problems, the use of symbolic representations is particularly well suited for situations in which the use of extra symbolic representations is often needed. ANNs and decision tree learning are both equally accurate when applied to this issue, and both may be used to get correct results in this case. The backpropagation algorithm is the most extensively used kind of program methodology for An artificial neural network (ANN) is a kind of computer network that uses artificial neural networks to perform tasks learning since it is the most efficient way available. The following are only a few examples of the countless themes for which it is appropriate:

- In order to represent instances, a large number of value pairs are utilized. The learning target function is described across instances that may be characterized by a vector of preset properties, such as pixel values, which are specified in the learning target function. Depending on the input qualities, there may be a strong connection or no connection at all. Any valid real value may be used as an input value, regardless of its type.
- The output of the function that is being targeted might be discrete or real-valued, or a vector of several discrete or real-valued variable properties.
- There may be inaccuracies in the training examples. Methods based on ANNs are very resistant due to the presence of noise in the training data
- Protracted training periods are permissible. Typically, training in the network methods need more time to train than, for instance, decision tree learning techniques. Training timeframes may vary Based on variables such as the amount of weights in the network, the number of training sessions, and other parameters samples reviewed, as

well as how different learning algorithms work settings are tuned, it might take anywhere from a few seconds to many hours to complete.

- Rapid assessment of the goal function for learning may be necessary. While ANN learning durations are lengthy, When it comes to studying and establishing a learning network in order to apply it to a future instance, time is usually on our side.
- Humans' capacity to comprehend the role of the learning objective is irrelevant. The weights learnt via use of neural networks are often troublesome to comprehend by humans. Humans have a harder time communicating with taught neural networks than they do with learned rules.

2.5. MLP

A network is composed of three levels: There are three levels in the system: an input layer, one or more hidden layers, and an output layer. An input layer is a layer that receives data at the lowest level of a network. The input is propagated across the network layer by layer. MLPs is an abbreviation for these types of neural networks. Until the mid-1980s, artificial neural networks (ANNs) were largely investigated in conjunction with multilayer networks, using the error backpropagation (EBP) learning method. EBP is a two-stage technique that traverses the network's various layers: a forward pass and a backward pass. It is used to detect and repair network problems. The forward pass is used as an input vector to the network's nodes, and the impact of the input vector propagates layer by layer throughout the network while the forward pass continues. Finally, a collection of outputs is generated that accurately mimics the real-time network reaction. All of the weights have been completely determined throughout this time period. During the reversal of direction, all weights are modified in line with the error correction rule to ensure that they are as accurate as possible. In order to transmit an error signal, this is accomplished by subtracting the actual network response from the projected network response. Due to the fact that this mistake propagates backward across theSynaptic connections flow in the opposite direction of the network, it is referred to as EBP. By modifying the synaptic weights, the actual network response may be brought closer to that anticipated by the model.

MLP networks include at least three layers: an input layer, one or more hidden layers, and an output layer, to name a few examples. An MLP network has at least three layers. In the real world, the nodes are linked by connections, which are represented by weights. Each node gets a huge number of input values, processes them, and outputs a value that other nodes may use to communicate with one another. The output of a node j is analogous to the following:

$$O_j = transfer\left(\sum X_{ji} W_{JI}\right) \quad (5)$$

O_j denotes the output of node j,Ith input to unit j is denoted by W_ji, which denotes the high association with the ith input to unit j, and ith input to unit j is denoted by X_{ji} transfer, which represents the nonlinear transfer functions responsible for transforming the weighted sum of the input to a value that is delivered to the next node. Despite the fact that they are linked in a network, a neuron might have an infinite number of inputs but only one output, indicating that they are not connected in a network. The weights of the connections connecting the nodes may be changed in order to approximate a given function, and the ANN can be changed in order to approximate a certain function.

2.6. Learning Algorithm

To guarantee that an ANN can approach the goal function with precision, it is common for the weights of an ANN to be adjusted via the use of a learning method. The stochastic gradient descent backpropagation learning approach will

be covered in depth in this portion of the book. In this context, An MLP is referred to as a "neural network" in the context of this document that has been trained using this approach, which is also known as backpropagation or EBP in certain circles. Take for example the case when an artificial neural network makes use of the error function (ANN).

$$E(\overline{W}) = \frac{1}{2} \sum_{d \epsilon D k \epsilon outputs} \sum (t_{kd} - O_{kd})^{2} \quad (6)$$

O_{kd} signifies the value of the output generated K is the neuron that produces the output, t_{kd} denotes the intended D is the (right) value that this neuron should generate, and A is the value that this neuron should produce collection in the case of all training regimens ; that is, $E(\overline{w})$reflects the total of the prediction errors for each training sample. When the prediction errors of each individual training example are added up together, they provide a total of the discrepancies between the output values generated by an ANN and the intended (right) values, where w is a vector representing a set of weights for an ANN.

D denotes the collection of all training patterns, and O_{kd} denotes the value generated by output neuron K, tkd denotes the intended (correct) value that this neuron should produce, and O_{kd} denotes the value generated by output neuron K. $E(\overline{w})$denotes the total of all prediction errors for all training instances combined There are a total of n training instances. The prediction errors associated with each training instance are equal to the sum of all the disparities between the ANN's output values and the target (correct) values, where w denotes the weight vector of the ANN and n denotes the number of training instances.

$E(\overline{w})$is the aim of a learning algorithm for a given collection of training samples, and the The purpose of a learning algorithm is to reduce the amount of data collected $E(\overline{w})$. It is possible to do this by a variety of approaches, one of which is the so-called gradient descent technique, which acts in the following way:

- Make a list of possible starting points for the model's parameters that have been (randomly) selected.
- Using the gradient G of the error function with respect to each model parameter, compute the gradient G of the error function error function.
- Create adjusting the settings in such a way that we only go a short distance in the direction where we may expect to see the greatest rate of error reduction, which is to say, in the direction of the letter G.
- Steps 2 and 3 should be repeated until G is close to zero.

Let G=$\Delta f(x)$, , the inverse of the function's gradient f ,be the vector containing the first partial derivatives of the function,

$$\Delta f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, .., \frac{\partial f(x)}{\partial x_n} \right) \quad (7)$$

$G = \Delta E(\overline{w})$ is used in our scenario For example, the derivative of the error function E with respect to the weight vector w is defined as The gradient descent backpropagation (EBP) approach for machine learning will be investigated in the next section with this in mind. The parameters of a neural network are set up as a starting point once it has been built (the weights are set to small random numbers). Afterwards, all training instances are instructed on how to utilize the ANN until the termination condition has been satisfied (e.g., the mean squared error of the ANN is less than a specified error threshold). When the inputs from each training sample are submitted to the ANN, it processes them from the input layer through the hidden layer(s) and out to the output layer in the order in which they were received. ANN output values are represented as a vector as a consequence of this procedure.

ISSN 2723-6471

The next step is to alter the ANN's weightings in some way. This is accomplished, in essence, by changing the weight in the direction of the error function's greatest decrease, as shown in the figure. In order to do this, the following value is added to each individual weight:

$$\Delta W = \eta \partial_{jX_{JI}} \quad (8)$$

where η is the pace of learning that affects the size of the step that we take in order to advance toward the minimum of E, as well as δ_j represents the neuron j's erroneous term in the equation The lengths of the arrows represent the pace at which the learner is progressing.

Numerous changes to this technique, including the inclusion of momentum terms and the decay of weight, have been described in the relevant literature. Nonetheless, in reality, MLPs combined with The most often used learning algorithms are stochastic gradient descent learning artificial neural networks. The fact that this approach is based on a quadratic error function is another important point to consider. This shows that There is just one minimal requirement. In practice, the error function is referred to as may deviate from the global minimum owing to the occurrence of several local minima in the error function. In certain cases, it is conceivable that the algorithm may end up in one of the minima on a local scale. It will be unable to eliminate mistakes to the greatest extent possible, the greatest degree possible via the use of a global minimum. An ANN design for our data is described in the next section, which is then compared step by step to a GARCH model.

3. Method

This section technique is based on Aslam et al. [5], The creation of a neural network capable of accurately forecasting a financial time series is a difficult problem. The following figures 1 are the various phases in this process.

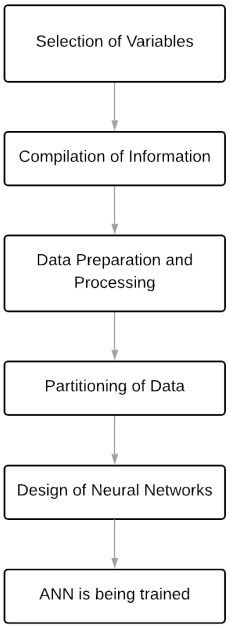


Fig. 1. Research Process

A thorough explanation of each step is provided further down on this page.

3.1. Selection of Variables

It is essential to have a thorough grasp of the situation at hand in order for neural network design to be successful. It is vital to identify the input factors that have a significant influence on the predicted market's performance. When it comes to using a neural network, this is easier said than done, considering that the fundamental aim for doing so is to discover complicated nonlinear connections between a huge number of distinct variables. However, economic theory may be useful in identifying components that are likely to be significant predictors of future outcomes. When it comes to this step of the design process, the raw data that will be used to generate a range of indications is given the most attention possible. These signals will be created by utilizing the real inputs to the neural network as inputs.

When forecasting stock market values, a financial researcher must decide whether to include both technical and fundamental economic data from a single market or from a number of other markets. Depending on the scenario, Technical inputs are provided as the lagged values of the dependent variable or as inferences taken from the lagged values of the dependent variable, depending on the situation. Using delayed values for the dependent variable variables that were obtained through correlation and autocorrelation analysis of the data, the model presented in this article can be constructed. Shou et al. [11] previously reported the sample autocorrelation functions (ACF) and partial autocorrelation functions (PACF) for each series, which we present here. In the next step, we'll look at the lags, as well as their accompanying ACF and PACF values, to see whether ones are connected with a statistically significant correlation. According to the previous section, the next step is that network delays should be applied to all of the chosen lags in a single operation. The number of delays in each series is kept to a bare minimum, and models are chosen using the conventional Bayesian information criterion (BIC) approach, which is based on the root mean square error (RMSE) (RMSE). To put it another way, correlation analysis and the BIC approach are utilized to determine the input variables. Table 2 lists the data structures that were utilized as inputs for the algorithms.

Table. 2. Variable selection

Variables	Past Values of the Closing
DOW	$DOW_{t-1}, DOW_{t-2}, DOW_{t-3}$
S&P	$S\&P_{t-1}, S\&P_{t-2}$
DAX	DAX_{t-1}, DAX_{t-2}
CAC	$CAC_{t-1}, CAC_{t-2}, CAC_{t-3}, CAC_{t-4}$
FTSE	$FTSE_{t-1}, FTSE_{t-2}, FTSE_{t-3}$
IBEX	$IBEX_{t-1}, IBEX_{t-2}$
PSI	$PSI_{t-1}, PSI_{t-2}, PSI_{t-3}, PSI_{t-4}$
IBOV	$IBOV_{t-1}, IBOV_{t-2}, IBOV_{t-3}$

The frequency with which data is collected is dictated by the objectives of the researcher. A typical off-floor trader in the stock or commodities futures markets would very certainly employ daily data to train a neural network as part of a larger trading strategy, rather than relying on historical data alone. Instead of using a passive buy-and-hold strategy, an investor with a longer time horizon may elect to use weekly or monthly data as inputs to a neural network in order to find the ideal asset allocation strategy.

3.2. Compilation of Information

Before beginning the process of collecting information on any of the factors specified in Step 1, the researcher must take the cost and availability of data into consideration. The availability of technical data at a reasonable price from a variety of sources is easily accessible; nevertheless, vital information is more difficult to get by. During the time spent gathering data, there is no time available for preparation, network training, or performance evaluation. However, although the The vendor should have a good reputation in the industry supplying high-quality data, and all data should be of high quality thoroughly reviewed for faults, including daily changes, ranges, logical consistency, and missing observations, before being used. Many different approaches may be used to address observations that are often neglected. Alternatively, missing data might be deleted entirely, or values can be interpolated or averaged from the data in the surrounding area, providing that the missing observations do not change. We make the assumption in this study that due to the fact that we want to depict stock markets, we must ensure that there are no missing observations in the sample and that specific values in the data that may be regarded as outliers exist in the data mostly during moments of turbulence with little losses.

3.3. Data Processing

In order to achieve high prediction accuracy while utilizing neural networks to forecast financial time series, it is essential to have a well-designed data processing system, as has been shown in a broad variety of other applications using neural networks. Although it is very unusual, the input and output variables used in data collection are not always connected directly to the network. In an ideal situation, the raw data should be scaled between the upper and lower bounds of the transfer function, however this is not always the case in practice (Usually between zero and one minus one and one, or between zero and one.). When it comes to classical and neural network forecasting, the initial difference of a variable and the logarithm of a variable are two of the most often utilized data transformations. The first step is to delete a linear trend from data by utilizing differencing, also known as variable modification, which is a statistical technique. When working with data that comprises both little and large numbers, the logarithmic transformation may be quite beneficial in some situations. In addition, Logarithmic transformations turn multiplicative or ratio interactions into additive connections, which are then converted back into multiplicative or ratio interactions simplifies and improves the training of neural network topologies by reducing the number of interactions. Consider the following scenario as an illustration of logarithmic modification of the return:

$$R_t = In\left(\frac{Index_t}{Index_{t-1}}\right) \tag{9}$$

In this case, R_t signifies the normal logarithm of the returns. According to the literature, in this context. Additionally, a normal probability distribution more closely approximates the behavior of This technique is very helpful for financial time series research, and it yields consistently strong results in this area as well the returns, although, as will be shown later, testing this hypothesis is a very challenging prospect.

3.4. Data Partitioning

Time series are often classified into three categories: training, testing, and validation. This categorization is widely used in the field (out-of-sample). The training set is the most extensive collection of data available, and it is used by neural networks in order to detect patterns in the data they encounter in their subsequent training. In order to examine the generalization capabilities of a network that has been allegedly trained in the first instance, a testing set that is between 10 and 30 percent the size of a training set is employed. Additionally, the validation data Obtaining a sufficient sample size to assess a trained network while keeping costs down is a delicate balance to achieve while still maintaining a reasonable number of residual Both training and testing applications make use of observation. This is because prior data were used to train the network, and since the most recent observations serve as a validation sample in order to verify whether or not the model is generalizable. As a result, the validation sample should be made of the most recent contiguous data. In normal time series models, the most recent observations serve as a validation sample, in a manner similar to how the most recent observations act as a validation sample. They are classified into the following categories, which are used to categorize and organize the sets:

1) 75% of the training set
2) 15% of the testing set
3) 10% of the validation set

3.5. Design of Neural Networks

A neural network may be built in an almost limitless number of different ways. The terms "neurodynamics" and "architecture" relate to the way a neural network is organized and how it functions. After the control variables have already been filtered, one of the simplest things to determine is the number of data neurons, due to the fact that each independent variable is represented by a distinct input neuron The calculation of the number of hidden layers, the number of neurons in the hidden layers, the number of input neurons, and the transfer functions, on the other hand, is far more difficult to do.

3.5.1. Hidden Layers

The network's capacity to generalize is measured in terms of is enabled through the use of hidden layers. Indeed, Neural networks with one or two hidden layers are considered to be extensively employed, and their performance has been shown to be better. As the number of hidden layers grows, so does the amount of time it takes to compute and the risk of overfitting, resulting in poor out-of-sample forecasting accuracy. It is determined by the amount of weights in neural networks, which is strongly related when it comes to the amount of hidden layers and neurons, as well as the quantity of the training data, which determines the likelihood of overfitting (number of observations). In this essay, we looked at the structure of neural networks with one or two hidden layers (one and two hidden layers), as well as the differences between them.

3.5.2. Hidden Neurons

Despite its significance, there is no secret formula for picking the optimum amount of buried neurons, and researchers have come up with a variety of different approaches that must thus depend on trial and error to determine the optimal number of hidden neurons. Certain generalizations, on the other hand, have been made. The geometric pyramid method described by Dbouk and Kryzanowski [19] may be used to derive an approximate estimate of the optimal number of hidden neurons in a given situation. The buried layer would contain n input neurons and m output neurons in Using n input neurons and m output neurons in a three-layer neural network. $\sqrt{n \, . \, m}$ neurons. Dinc and

Kryzanowski [20] suggest that the number of hidden layer neurons in a three-layer neural network should be 75% of the number of input neurons. Black et al., [21] indicates that the optimal number of hidden neurons will generally be found between one-half to three times the number of input neurons. He [22] proposes doubling the number of hidden neurons until the network's performance on the testing set deteriorates. For example, according to Manickavasagam [23], there should be at least five times the number of training facts as weights, which provides an upper limit on the total number of input and neurons. To test for these characteristics, this study applies multiple structures to all of the data, picking them at random, utilizing 2, 3, 4, 5, and 6 neurons in the hidden layer to discover the optimal structure based on the index, and then examines the outcomes.

3.5.3. Output Neurons

BecauseThere are compelling arguments in favor of always using a single output neuron, determining the number of neurons to use is a little less challenging. If you compare the results of multiple output neural networks to those of neural networks with a single output, the results are much worse, particularly when the outputs are widely separated. When creating the output layer for this experiment, a network with a single neuron was used. The single The price of the closing price is represented by the neuron one step ahead of the current price.

3.5.4. Transfer Function

The sigmoid transfer function is the most often employed transfer function in contemporary neural network models; however, other transfer functions, such as the hyperbolic tangent, the arc tangent, and the linear transfer function, have been proposed as potential replacements. It is necessary to utilize linear transfer functions because of the nonlinear nature of mapping and classification. Nonlinear transfer functions, according to Jarrett [13-14], are more suited for financial markets since they are both nonlinear and memory-based [13-14]. For time series data, nonlinear transfer functions like the sigmoid are often utilized because of their continuous differentiability and nonlinearity, both of which are important qualities in network learning. The sigmoid transfer function is used in this research in order to construct the network that is proposed.

3.6. Training the ANN

In order to train the use of a neural network to recognize patterns in data, it is required to expose it to samples of correct known responses over and over again over an extended period of time. With training, the aim becomes identifying a set of weights that will result in the lowest feasible error function throughout the whole globe. This weighting technique should be sufficient for generalization in the vast majority of instances, unless the model is very well-fit. The gradient descent method is specifically employed to train the backpropagation network in this research; this technique automatically alters the weights to descend the As the network is trained, the sharpest slope of the error surface is reached. There are no assurances that the global minimum will be found, since the error surface may include numerous local minima that the algorithm may miss and may have difficulties traversing, making it difficult to identify the global minimum. As well as addressing when it is acceptable to terminate the training of a neural network, this section also discusses the selection of the learning rate and momentum parameters for the neural network.

3.6.1. Training Iterations

According to a significant number of studies that have looked at the When the number of training iterations is between 85 and 5,000, convergence occurs in rounds of training, depending on the study in question. This is in contrast to the fact that between 50,000 and 191,400 repetitions, as well as 60-hour training sessions, have all been

described in the literature, demonstrating that the range is rather large. The training process is affected by a variety of parameters, including the selection of the learning rate and momentum values, as well as tweaks to the unique backpropagation algorithm utilized by the system. Because these variables vary from research to study, it is impossible to set a standard figure for the maximum number of runs that should be carried out in a given situation.

On top of all that, the The software for neural networks has a high level of numerical accuracy can have an impact on training because, in some cases, theIt is possible for the slope of the error derivative to become incredibly tiny, resulting in Some neural network programs are going in the opposite direction of their intended orientation as a result of round-off errors that accumulate quickly in the highly iterative training algorithm. The software for neural networks has a high level of numerical accuracy It is advised that researchers create their own unique challenge and experiment with as many random beginning weights as their computing restrictions will allow them to do so. In order to find out which index performs the best, we will employ a random selection of iterations with values ranging from 100 to 250 to 500 to 800 and 1,200.

3.6.2. Learning Rate

This is referred to as an excessive learning rate, and it occurs during the training process if the error function changes drastically without improving further. Additionally, a slow pace of learning demands the need for extra training hours. It is necessary to vary the learning rate during training or to brainwash the network by assigning all weights at random and changing the learning rate for each successive trip through the training set in order to ensure that the network learns at its optimal rate during training or to brainwash the network. To ensure that the network learns at its optimal rate during training or to brainwash the network, it is necessary to vary the Increasing the learning rate during training or brainwashing the network are two options. In this research, the initial learning rates used varied from 0.1 to 0.9, depending on the task under consideration. The default learning rate parameters are pre-configured in the overwhelming majority of neural network software packages, and these values are found to be effective in the vast majority of application scenarios. In the industry, it is standard practice to start training with a higher learning rate, such as 0.7, and then progressively reduce it as the training develops. Many network algorithms have their learning rate lowered automatically as they approach convergence, ensuring that the algorithm achieves its target successfully in the end.

4. Results and Discussion

This section covers the neural network structure that was utilized to create the data with the fewest number of errors as feasible, and how it was achieved. In addition, the results of the ANN and GARCH models are provided for the purpose of comparison and contrast. Table 3 summarizes the results of each index analyzed, as well as the optimal neural network architecture for each index assessed.

Table. 3. Design of neural networks

Indicator	Inputs	Hidden Layer(s)	Hidden Neurons	Iterations	Learning Rate
DOW	4	3	5	850	0.5
S&P	3	3	7	550	0.6
DAX	3	2	3	850	0.5
CAC	4	3	4	1250	0.6

FTSE	4	2	3	550	0.8
IBEX	3	2	6	850	0.6
PSI	5	3	3	300	0.7
IBOV	4	3	4	850	0.6

The findings indicate that the structure chosen varies according to the data. There is no one-size-fits-all method for describing a structure that minimizes mistakes and produces the optimum outcome. According to the facts, the optimal decision must be determined using random alternatives.

Because it produces less mistakes than GARCH models, it was determined during the trials that the suggested method is a feasible alternative to GARCH models for stock market forecasting (see Table 4). Using the ranking multiple determination coefficients for each model, the results in Table 4 show which model is the most accurate based on its ranking. The R-squared value, which indicates what proportion of the variance in the dependent variable can be explained by the independent variables, is used to depict this relationship. The R-squared value of the model must increase in order for it to be more efficient at explaining variance in the dependent variable. Without additional comparison, it is shown that the neural network is the most efficient approach of accounting for dependent variable fluctuation, followed by the regression model, in this case. For the sake of comparison, the ranking error numbers have been included as well. The difference between the goal value and the output value of the neural network is utilized to produce all of these statistics, which are all reliant on this difference.

Table. 4. Error comparison

Index	R Squared		Percentage Mean Error		RMSE		POCID	
	ANN	GARCH	ANN	GARCH	ANN	GARCH	ANN	GARCH
DOW	0.96336	0.85337	3.88333	7.63427	0.63153	2.84223	75.68%	64.76%
S&P	0.94442	0.72439	2.72283	6.79330	0.88324	3.84283	76.24%	58.79%
DAX	0.97742	0.86374	4.97331	8.68384	0.64264	2.72328	73.96%	59.91%
CAC	0.95337	0.82282	3.02943	7.42654	0.94290	4.03392	75.54%	62.14%
FTSE	0.96352	0.78332	4.31197	6.53734	0.74733	3.94812	78.05%	56.25%
IBEX	0.90773	0.85352	3.08333	7.43724	0.84222	2.84918	79.66%	60.98%
PSI	0.92731	0.77863	2.66337	6.58431	1.84284	5.64262	70.04%	50.65%
IBOV	0.95380	0.79333	2.04125	9.63922	0.64284	3.64784	79.12%	63.12%

On the basis of a visual evaluation, the neural network model outperforms the GARCH process, as shown in Table 4. It is important to distinguish between this and the model ranking, which is determined by the R-squared values of the models. It is possible to anticipate the closing value with a high degree of accuracy when using a neural network, as

in this model. As part of our efforts to estimate how long the ANN model will last, we are looking into the error dimension in the sets that have been finished (training, test, and validation). The maximum percent error (MPE) and root mean square error (RMSE) metrics are used to quantify the results, which are as follows:

$$MPE = max\left\{\frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - y_i}{y_i}\right|\right\} \quad (10)$$

$$RMSE = \sqrt{\frac{1}{n}(y_i - y_i)^2} \quad (11)$$

Here, y_i represents the desire value i, y_i and the output of the neural network, respectively. The financial market is a dynamic system that is complex, evolving, and nonlinear in nature. In the world of finance, a variety of elements: Political changes, general economic conditions, and the expectations of traders are all factors that interact with one another. As a consequence, anticipating variations in financial prices is very difficult, but it is also quite crucial. Using another essential evaluation metric, the precision of direction prediction accuracy (POCID), we may describe the accuracy of prediction of change in direction as follows:

$$POSID = 100\frac{\sum_{i-1}^{N} D_i}{n} \quad (12)$$

Where $D_i = 1$ if $(y_i - y_{i-1})(\widehat{y_i} - \widehat{y_{i-1}}) > 0$, or $D_i = 0$, otherwise.In this scenario, the model with a higher POCID is more accurate at forecasting the moves of the stock market. The network sets are compared in Table 5.

Table. 5. Network set comparison

Index	Sets								
	Training			Test			Validation		
	MPE	RMSE	POCID	MPE	RMSE	POCID	MPE	RMSE	POCID
DOW	4.42713	2.22531	75.17%	4.97136	2.97531	74.12%	4.81284	3.23144	71.15%
S&P	7.63273	3.52523	74.93%	8.02187	3.97542	77.54%	8.18653	4.15245	74.99%
DAX	3.44742	2.46292	77.89%	3.82372	2.86522	74.27%	4.19357	3.42722	72.42%
CAC	6.93213	3.88246	76.60%	7.63142	4.23273	71.20%	7.25134	4.83522	70.20%
FTSE	6.07273	3.18231	80.02%	6.12193	4.12148	79.64%	6.58734	4.63299	77.24%
IBEX	6.84163	2.31352	73.12%	7.31643	3.52166	73.00%	7.43291	4.12581	71.87%
PSI	6.36325	4.86542	69.95%	6.93645	5.83522	68.98%	7.11973	6.10022	69.18%
IBOV	4.63128	3.19227	73.14%	5.12756	4.23841	71.11%	5.11466	4.73228	72.09%

Because of its high predictive capability, as demonstrated by the The neural network structure applied to all indexes has a low rate of validation errors when compared to the GARCH model (see Table 4), demonstrating that it is capable of learning from data and processing accurate forecasting results. This is demonstrated by the low rate of

validation errors when compared to the GARCH model (see Table 4). Comparing the ANN model to the GARCH model, which has a larger error rate than the ANN model, it becomes clear that neural networks have superior performance. According to the results of theThe neural network is capable of generalization, as shown by the test and validation sets , while Despite the fact that the differences in POCID values between the ANN and GARCH models are not statistically significant, they do imply that the ANN model deployed anticipates market changes more accurately than the GARCH model (see Tables 4 and 5). The GARCH model correctly predicts an average of 60 percent however, the neural network model does not account for market price changes (either up or down) reliably anticipates this measure to an extent of around 75 percent, as shown in Table 4. ANNs are widely recognized for producing low POCID values when compared to the other sets in the validation set, and this has been confirmed in several studies.

This study quantifies prediction performance for all indices using conventional statistical loss, an approach frequently utilized by others, including Aslam et al. [5], but these tests do not reveal if one model's forecast is statistically superior to another's. As a consequence, another commonly used test is necessary to help in the evaluation of the prediction accuracy of competing models amongst themselves. Few papers in the literature utilize statistical tests to verify competing models, and when they do, it is only when one of the models is a neural network model that the statistical tests are used. As a consequence, in addition to the normal metrics outlined above, we conduct two parametric tests of paired prediction assessments. The results of these tests are given below.

This research makes use of the AGS test, which was designed by Loo [11] and is one of the parametric tests that were employed. To establish the statistical significance of the difference between two competing model forecasts, it is necessary to compare their root mean square errors (RMSEs). The following equation serves as the foundation for our investigation:

$$d_t = \beta_1 + \beta_2\left(S_t - S_{mean}\right) + \zeta_t \tag{13}$$

where dt is the difference between the prediction errors produced by the ANN and GARCH models, t s is the total of these forecast mistakes, mean s is the sample mean of t s, and t s is the time interval between the forecast errors acquired by the ANN and GARCH models ζ is a procedure characterized by white noise In computer science, the AGS test is a comparison of the mean squared errors (MSEs) of two competing models against the alternative that the second model's MSE is lower (and so more accurate) than the first model's. The test is carried out by examining the relevance of the parameters in conjunction with one another 1β and 2 β in regression (13). The residuals produced from estimating an unconstrained model represented by equation (13) and a model that limits the model represented by equation (14) are used to construct the test statistics for the AGS test β1 = β2 = 0 in equation (13). The following equation is used to determine the test statistics for this particular examination:

When the difference between the prediction errors gained by ANN and GARCH models is taken into account, the total forecast mistakes are taken into consideration. The mean of the sample mean of the forecast errors is taken into consideration, as is the sample mean of the ANN model ζ The white noise technique is referred to as When two competing models' mean squared errors (MSEs) are compared, the AGS test determines if the second model's MSE is smaller (and hence more accurate) than the first model. The examination of the relevance of the characteristics is carried out during the test 1β and 2β in the regression equation when they are used together (13). With the residuals acquired from the estimate of an unconstrained model represented by equation (13) and a model that restricts 1 = 0 = 0 in equation (13) respectively, it is able to build the test statistics for the AGS test (13). When determining the test statistics for this specific examination, the following equation is used:

ISSN 2723-6471

$$TS = \left(\left(SSE_R - SSE_{UR}\right)/(k-1)\right)/\left(SSE_{UR}/(n-k)\right) \quad (14)$$

An example of a multivariate regression model is one in which there are several variables in the regression model. An example of a multivariate regression model is one in which there are several variables in the regression model. Assuming that the prediction errors are not contemporaneously related with one another and that they are not susceptible to serial correlations, this test may be performed with confidence. Under the assumption of normality, the test statistic for this test is distributed F with two and n – two degrees of freedom, respectively, according to the normal distribution. This is yet another parametric test that evaluates forecast accuracy on an equal footing with other tests of accuracy. When calculating the test statistics for this test, the following equation may be utilized:

$$MGN = \frac{\widehat{P}_{sd}}{\left(1-\widehat{P}_{sd}\right)^{12}}(n-1)^{12} \quad (15)$$

Model 1 errors are represented by e1 and model 2 errors are represented by $\widehat{P}_{sd}$, where r1 is the estimated correlation coefficient between the two models and r2 is the correlation coefficient between the two models, and Psd is the sum of model 1 and model 2 errors; e1 and e2 are the errors of model 1 and model 2, respectively. When it comes to this research, the GARCH model is represented by 1, and the ANN model is represented by 2. The probability distribution of the test statistics for the MGN test has n – 1 degrees of freedom, and it is denoted by the letter t. If all of the predictions for this test are equally right, If none of the predictions are equally correct, the correlation between s and d will be 0, and vice versa. Prediction accuracy metrics for the series are obtained using the AGS and MGN tests of forecast equivalence between two competing models, which are performed on the series data. This information is included in Table 6, as well as the forecast equivalency metrics for the series in question. ANNs outperform GARCHs in all of the indices studied in this work, according to AGS and MGN tests (which compare predictions generated by two competing models using ANNs and GARCHs as benchmarks). The ANN surpasses the GARCH in all indexes assessed in this study. In Table 6, we have statistically substantial evidence of higher performance in all of the measures assessed in this research, as evidenced by the data. ANN outperforms the GARCH model when it comes to anticipating future trends, according to all of the stock market indices studied in this study, which validates what has already been shown using traditional forecasting methodologies. Figures 1 through 8 provide graphical representations of the results. Figure 1: Specifically, we provide (a) the actual value of the index in combination with GARCH and ANN forecasts, and (b) for each series the difference between the genuine value of the index in conjunction with the outcomes of each model given as an error measure. In addition, we demonstrate (a) the genuine value of the index in combination with GARCH and ANN forecasts. There are no discrepancies between any of the results and what was discovered in the validation set.

Table. 6. AGS and MGN tests used for all of the indexes

Index	**Tests**			
	AGS Test		**MGN Test**	
	Statistic	P-Value	Statistic	P-Value
DOW	347.8864	(0.01)	1.9831	(0.01)
S&P	667.7356	(0.01)	-2.1457	(0.01)
DAX	512.1303	(0.01)	-1.2353	(0.02)

ISSN 2723-6471

CAC	599.7357	(0.01)	3.6846	(0.01)
FTSE	338.7336	(0.01)	-1.8764	(0.02)
IBEX	710.8464	(0.01)	-0.7866	(0.03)
PSI	457.1083	(0.01)	2.8473	(0.00)
IBOV	346.8827	(0.01)	-2.8468	(0.01)

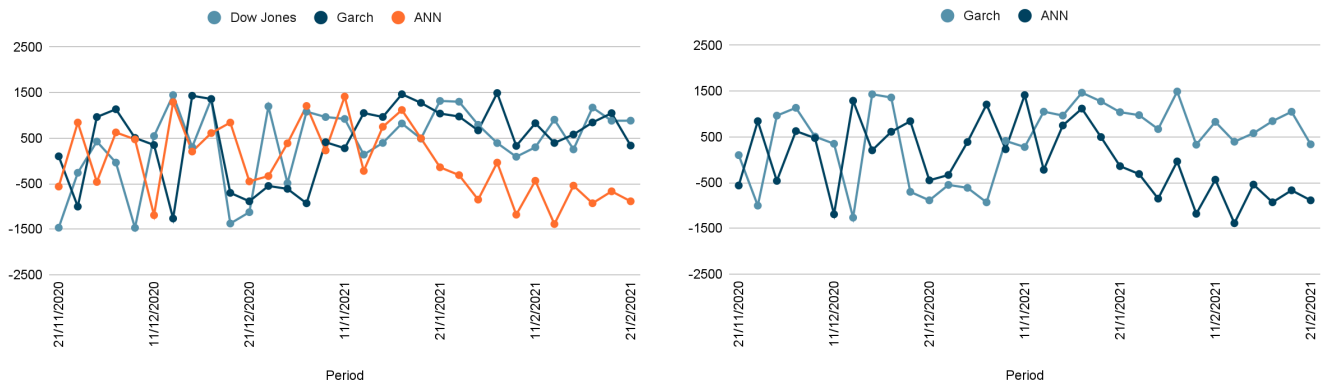


Figure. 2. Dow Jones Result

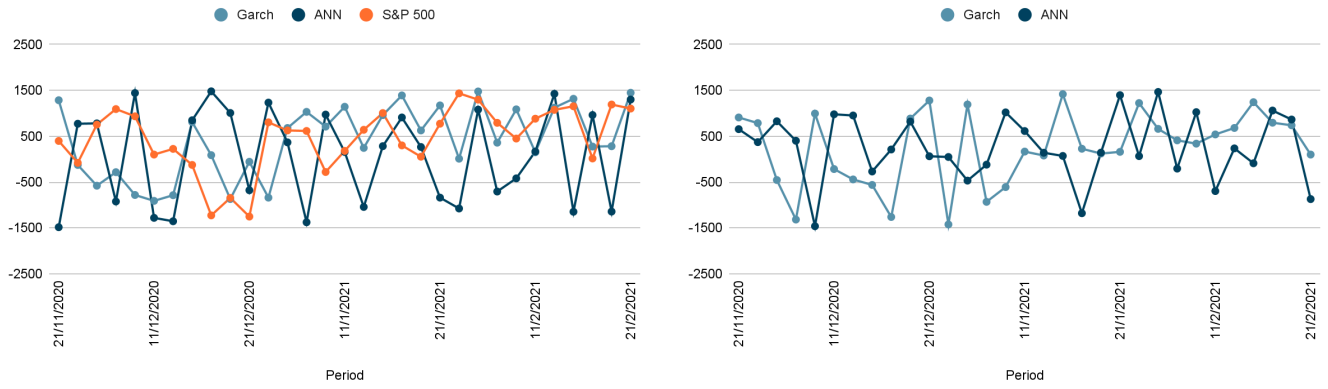


Figure. 3. S&P 500 Result

ISSN 2723-6471

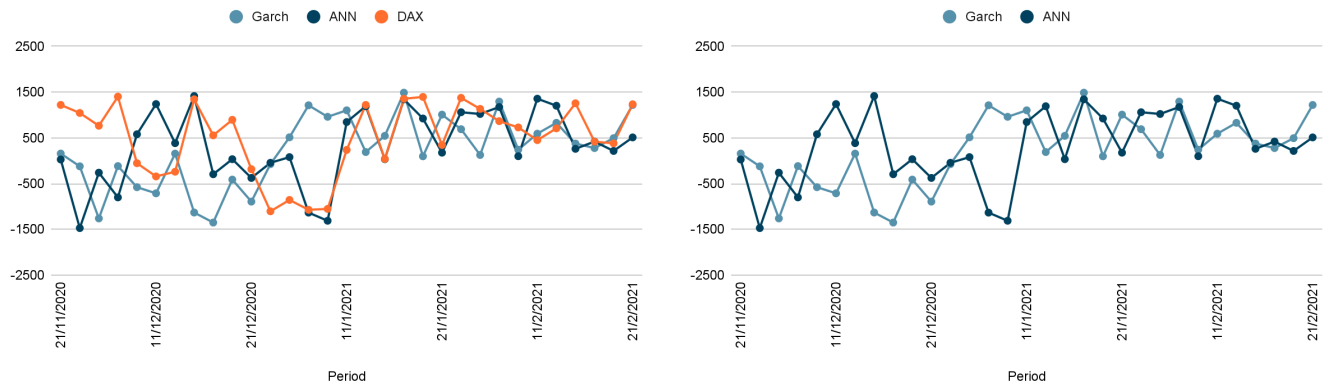


Figure. 4. DAX Result

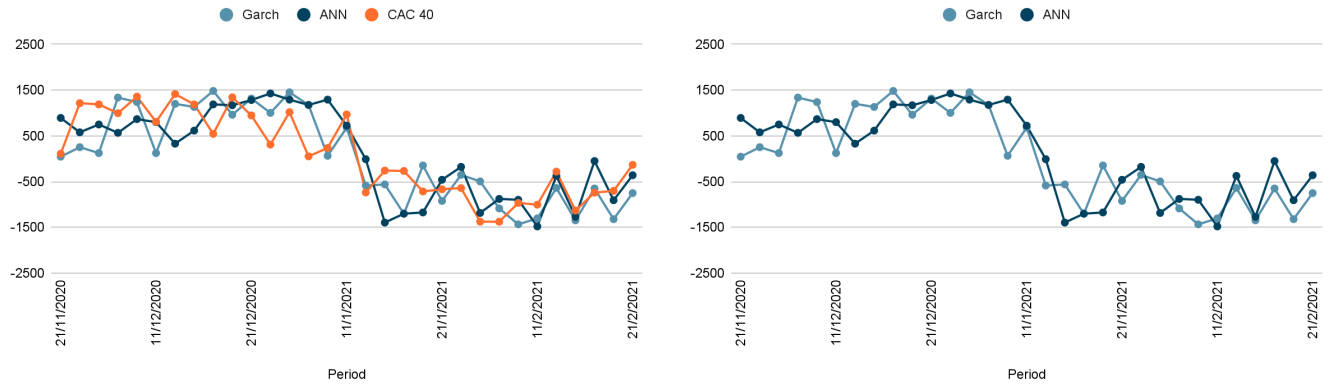


Figure. 5. CAC 40 Result

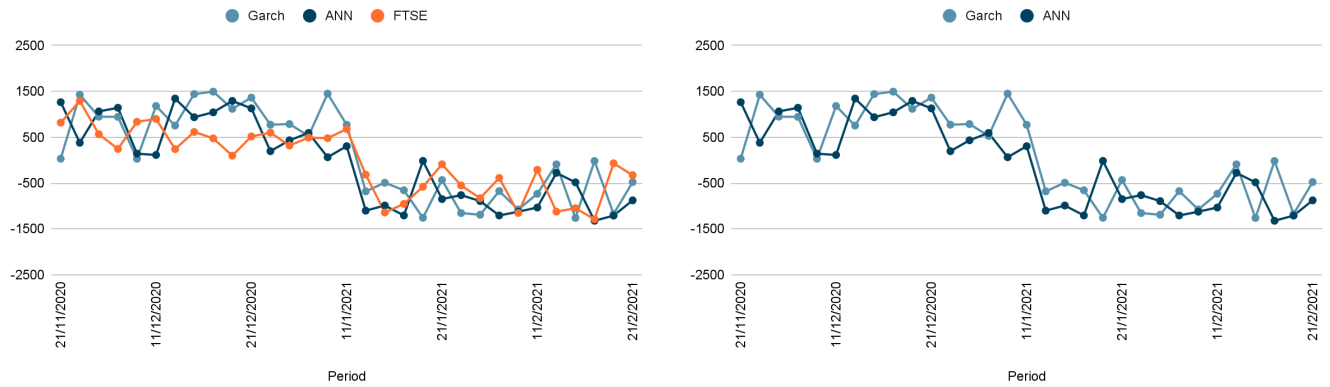


Figure. 6. FTSE Result

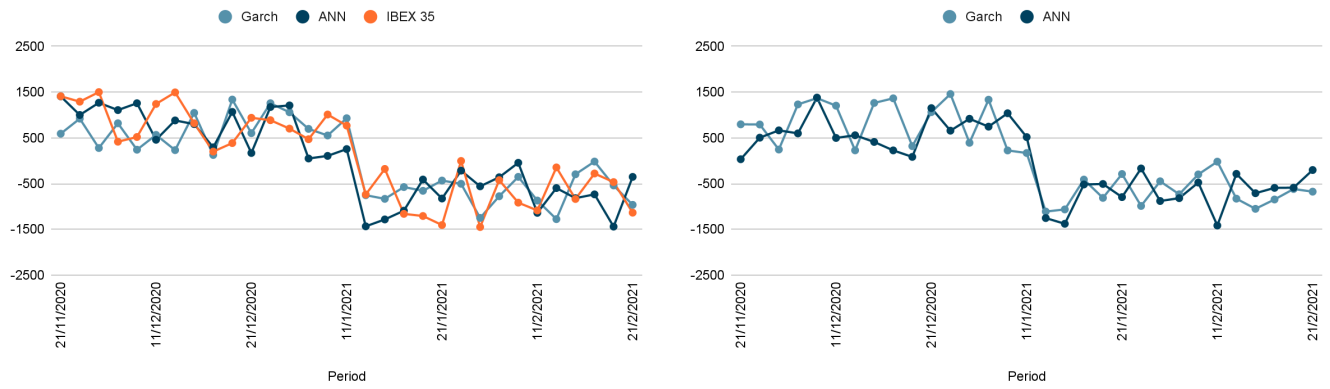


Figure. 7. IBEX 35 Result

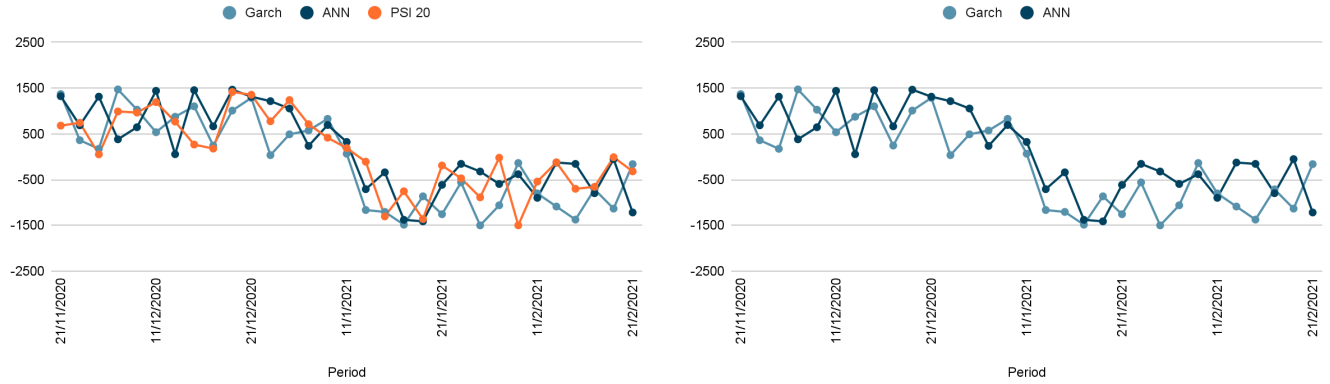


Figure. 8. PSI 20 Result

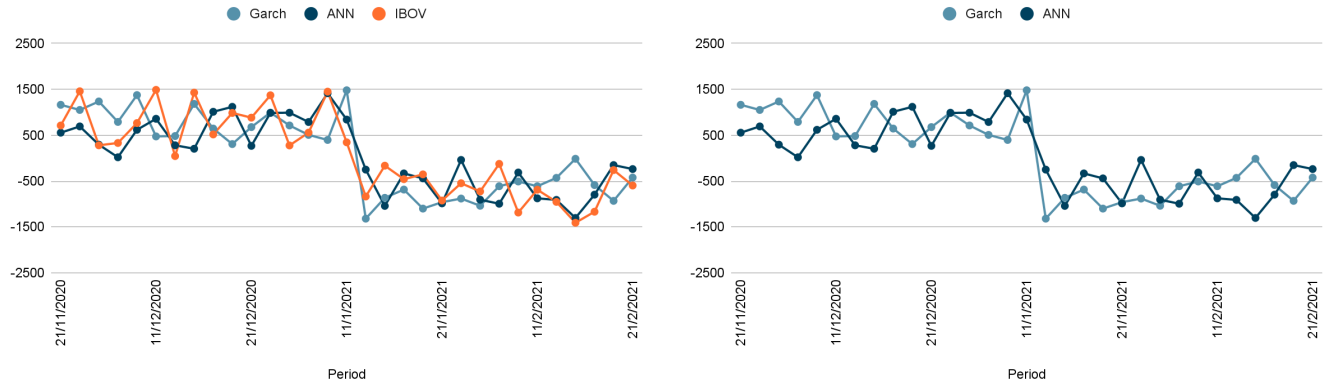


Figure. 9. IBOV Result

The conclusions presented in the previous section are supported by figures 2–9. The ANN technique outperforms the GARCH method in terms of accuracy in forecasting all of the indices evaluated. According to the discrepancy between actual and expected values obtained for each model, it can be concluded. Errors generated by the ANN are close to zero, but errors generated by the standard statistical model are much higher. In terms of error measurements, it has been shown that the intelligent artificial methodology is a successful way for predicting financial time series in

terms of error measurements. When it comes to neural networks, one of the questions posed in this research is whether or not they are capable of assimilating heteroskedastic occurrences. To summarize the residual analysis for each series covered in this paper, Summary of the findings are shown in Table 7. The mean test, normality testing (Jarque–Bera), a residual correlations test (the Ljung–Box–Pierce Q-test), and an ARCH test (Engle's ARCH test) to detect residual heteroskedasticity are all part of this set of procedures.

Table. 7. Residuals analysis

Index	Tests							
	Mean		Jarque-Bera		Ljung-Box-Pierce		Engle's ARCH	
	Value	Statistically Zero	Value	Normal	Value	Correlation	Value	Homoscedasticity
DOW	0.003	True	19.983	True	29.931	False	11.782	True
S&P	0.009	True	17.993	True	30.880	False	8.7891	True
DAX	0.089	True	56.934	True	31.822	False	12.785	True
CAC	0.058	True	62.993	True	26.791	False	9.7896	True
FTSE	0.069	True	26.908	True	27.842	False	13.972	True
IBEX	0.069	True	57.943	True	34.542	False	12.886	True
PSI	0.078	True	23.383	True	28.788	False	9.871	True
IBOV	0.049	True	55.872	True	32.882	False	11.881	True

As shown in Table 7, when we look at the neural network residuals for all of the indices examined, we can see that they that all of the residuals have the same homoscedastic distribution, that all of the residuals have a mean that is statistically equal to zero, and that all of the residuals have a normal distribution As a result, as shown by these data, the proposed neural network structure is capable of modeling and forecasting series, capturing heteroskedastic occurrences, and exhibiting the durability of the proposed approach.

5. Conclusion

It was the researchers' goal to look into the use of neural networks as a forecasting tool, and in particular, the evaluation of their capacity to anticipate future trends in the stock market indices. The researchers set out to look into the use of neural networks as forecasting tools in order to learn more about how they work. Stock market indices in North America, Europe, and Brazil were compared against the accuracy of a typical forecasting approach based on historical data to determine which was more accurate (GARCH). It is the purpose of this study to examine the Using neural networks as a forecasting tool for individuals has the potential to be both feasible and practical investors. There is just a basic introduction to neural network theory included in this research. In the conclusions of the study, individual investors may profit from the use of neural networks as an alternative to conventional forecasting methodologies for the following reasons. All of the stock market indexes analyzed show that neural networks are capable of forecasting them.

Multiple linear regressions need the accuracy of the underlying regression assumptions in order to be successful. The linearity and normal distribution assumptions may not hold According to the findings of current study, this is true for the vast majority of financial time series. The use of neural networks for nonlinear system modeling avoids making any assumptions about the probability distribution of the input data, which is particularly useful when dealing with large datasets. ANNs are approximate representations of universal functions that are used in mathematics. It has been shown that a neural network is capable of accurately estimating any continuous function with great precision. It is shown that ANNs are generalizable. Artificial neural networks (ANNs) are typically capable of correctly predicting the unseen component of a population after being trained on the data that has been supplied to them. This is true even when the sample data includes a high level of noise information According to standard measurement tests, neural networks outperform GARCH models in terms of accuracy, while ANNs surpass GARCH models in terms of statistical performance, as shown by the AGS and MGN tests. ANNs have the potential to record events that are heteroskedastic in nature.

Using neural networks in conjunction with other methodologies for financial and economic forecasting, Genetic algorithms,In addition to wavelet analysis, fuzzy inference, pattern recognition, and classic time series models, other techniques are also used, among other techniques, which will be the next step in future research efforts. The benefits of genetic approaches include their adaptability and resilience, as well as the fact that they do not have the problems associated with neural networks. being stuck at a local optimum, which may occur with other systems. Following the training, testing, and demonstration of effectiveness of a network, it may be improved using a genetic algorithm to improve its performance. Mutation and crossover are two mechanisms used by genetic evolution to modify the connections between neurons in a trained network. When it comes to financial data forecasting, hybrid systems seem to be well suited. On the other hand, depending on the dataset under consideration, the discussion of input variables may be construed in a different way.

References

[1] Y. Sun, Q. Jin, Q. Cheng, and K. Guo, “New tool for stock investment risk management,” Ind. Manag. Data Syst., vol. 120, no. 2, pp. 388–405, Jan. 2020, doi: 10.1108/IMDS-03-2019-0125.

[2] Y. Huang, C. Deng, X. Zhang, and Y. Bao, “Forecasting of stock price index using support vector regression with multivariate empirical mode decomposition,” J. Syst. Inf. Technol., vol. 2, no. 11, pp. 55–76, Jan. 2020, doi: 10.1108/JSIT-12-2019-0262.

[3] B. Liu and M. Tan, “Overconfidence and forecast accuracy,” Stud. Econ. Financ., vol. 38, no. 3, pp. 601–618, Jan. 2021, doi: 10.1108/SEF-12-2017-0345.

[4] Y. Zhang, J. Gao, and H. Zhou, “Breeds Classification with Deep Convolutional Neural Network,” ACM Int. Conf. Proceeding Ser., pp. 145–151, 2020, doi: 10.1145/3383972.3383975.

[5] F. Aslam, K. S. Mughal, A. Ali, and Y. T. Mohmand, “Forecasting Islamic securities index using artificial neural networks: performance evaluation of technical indicators,” J. Econ. Adm. Sci., vol. 37, no. 2, pp. 253–271, Jan. 2021, doi: 10.1108/JEAS-04-2020-0038.

[6] S. Rath, B. K. Sahu, and M. R. Nayak, “Application of quasi-oppositional symbiotic organisms search based extreme learning machine for stock market prediction,” Int. J. Intell. Comput. Cybern., vol. 12, no. 2, pp. 175–193, Jan. 2019, doi: 10.1108/IJICC-10-2018-0145.

[7] R. Östermark, “The forecasting performance of Cartesian ARIMA search and a vector‐valued state space model,” Kybernetes, vol. 29, no. 1, pp. 83–104, Jan. 2000, doi: 10.1108/03684920010308862.

[8] R. M. K. T. Rathnayaka, D. M. K. . Seneviratna, and W. Jianguo, “Grey system based novel approach for stock market forecasting,” Grey Syst. Theory Appl., vol. 5, no. 2, pp. 178–193, Jan. 2015, doi: 10.1108/GS-04-2015-0014.

[9] S. B. Liau and S. B. Reid, “Combined deep learning classifiers for stock market prediction: integrating stock price and news sentiments,” Kybernetes, vol. 4, no. 6, Jan. 2021, doi: 10.1108/K-06-2021-0457.

[10] Y. Seetharam and J. Britten, "Non-linear modelling of market cycles in South Africa," Int. J. Emerg. Mark., vol. 10, no. 4, pp. 670–683, Jan. 2015, doi: 10.1108/IJoEM-05-2013-0079.

[11] M.-H. Shou, Z.-X. Wang, D.-D. Li, and Y.-T. Zhou, "Forecasting the price trends of digital currency: a hybrid model integrating the stochastic index and grey Markov chain methods," Grey Syst. Theory Appl., vol. 11, no. 1, pp. 22–45, Jan. 2021, doi: 10.1108/GS-12-2019-0068.

[12] W. K. Loo, "Predictability of HK-REITs returns using artificial neural network," J. Prop. Invest. Financ., vol. 38, no. 4, pp. 291–307, Jan. 2020, doi: 10.1108/JPIF-07-2019-0090.

[13] J. E. Jarrett, "Random walk, capital market efficiency and predicting stock returns for Hong Kong Exchanges and Clearing Limited," Manag. Res. News, vol. 31, no. 2, pp. 142–148, Jan. 2008, doi: 10.1108/01409170810846858.

[14] J. E. Jarrett, "Efficient markets hypothesis and daily variation in small Pacific‐basin stock markets," Manag. Res. Rev., vol. 33, no. 12, pp. 1128–1139, Jan. 2010, doi: 10.1108/01409171011092185.

[15] R. Östermark, "Multiple input transfer function noise modelling in the time domain, Empirical evidence on Scandinavian stock data," Kybernetes, vol. 29, no. 3, pp. 355–380, Jan. 2000, doi: 10.1108/03684920010795312.

[16] R. Östermark, "Modelling dynamic systems with biased regression and spectral methods," Kybernetes, vol. 24, no. 6, pp. 38–43, Jan. 1995, doi: 10.1108/03684929510094271.

[17] P. Yin, G. Dou, X. Lin, and L. Liu, "A hybrid method for forecasting new product sales based on fuzzy clustering and deep learning," Kybernetes, vol. 49, no. 12, pp. 3099–3118, Jan. 2020, doi: 10.1108/K-10-2019-0688.

[18] S. P. Mohanty, S. Gopalkrishnan, and A. Mahendra, "The intertwined relationship of shadow banking and commercial banks' deposit growth: evidence from India," Int. J. Innov. Sci., vol. 3, no. 6, pp. 33–57, Jan. 2021, doi: 10.1108/IJIS-01-2021-0022.

[19] W. Dbouk and L. Kryzanowski, "Determinants of credit spread changes for the financial sector," Stud. Econ. Financ., vol. 27, no. 1, pp. 67–82, Jan. 2010, doi: 10.1108/10867371011022984.

[20] A. Dinc and A. Mamedov, "Optimization of surface quality and machining time in micro-milling of glass," Aircr. Eng. Aerosp. Technol., vol. 3, no. 6, Jan. 2021, doi: 10.1108/AEAT-06-2021-0187.

[21] A. J. Black, D. G. McMillan, and F. J. McMillan, "Cointegration between stock prices, dividends, output and consumption," Rev. Account. Financ., vol. 14, no. 1, pp. 81–103, Jan. 2015, doi: 10.1108/RAF-09-2013-0103.

[22] L. T. He, "Forecasting of housing stock returns and housing prices," J. Financ. Econ. Policy, vol. 7, no. 2, pp. 90–103, Jan. 2015, doi: 10.1108/JFEP-01-2014-0004.

[23] J. Manickavasagam, "An investigational analysis on forecasting intraday values," Benchmarking An Int. J., vol. 27, no. 2, pp. 592–605, Jan. 2020, doi: 10.1108/BIJ-11-2018-0361.