Journal of Computational Design and Engineering, 2023, **10**, 1110–1125
DOI: 10.1093/jcde/qwad039
Advance access publication date: 2 May 2023
Research Article

Hybrid neural network-based metaheuristics for prediction of financial markets: a case study on global gold market

Mobina Mousapour Mamoudan[1], Ali Ostadi[2], Nima Pourkhodabakhsh[3], Amir M. Fathollahi-Fard[4] and Faezeh Soleimani[5,*]

[1]School of Industrial and Systems Engineering, College of Engineering, University of Tehran, Tehran 11155-4563, Iran
[2]Student, Faculty of Information Systems and Applied Computer Sciences, University of Bamberg, 96047, Germany
[3]School of advanced sciences and technology, College of Management, Islamic Azad University of Medical Sciences, Tehran 1949635881, Iran
[4]Peter B. Gustavson School of Business, University of Victoria, 1700, Victoria, BC V8P5C2, Canada
[5]Department of Mathematical Sciences, Ball State University, Muncie, Indiana 47306, USA
*Corresponding author. E-mail: fsoleimani@bsu.edu

Abstract

Technical analysis indicators are popular tools in financial markets. These tools help investors to identify buy and sell signals with relatively large errors. The main goal of this study is to develop new practical methods to identify fake signals obtained from technical analysis indicators in the precious metals market. In this paper, we analyze these indicators in different ways based on the recorded signals for 10 months. The main novelty of this research is to propose hybrid neural network-based metaheuristic algorithms for analyzing them accurately while increasing the performance of the signals obtained from technical analysis indicators. We combine a convolutional neural network and a bidirectional gated recurrent unit whose hyperparameters are optimized using the firefly metaheuristic algorithm. To determine and select the most influential variables on the target variable, we use another successful recently developed metaheuristic, namely, the moth-flame optimization algorithm. Finally, we compare the performance of the proposed models with other state-of-the-art single and hybrid deep learning and machine learning methods from the literature. Finally, the main finding is that the proposed neural network-based metaheuristics can be useful as a decision support tool for investors to address and control the enormous uncertainties in the financial and precious metals markets.

Keywords: bidirectional gated recurrent unit, convolutional neural network, firefly algorithm, moth-flame optimization algorithm, precious metals market, prediction

1. Introduction

Access to developed financial markets is necessary for achieving sustainable and rapid economic growth in our modern world. These markets have always been the focus of economic activists and academics, as they are one of the primary sources of financial resources, economic growth, and development in any economy. The opportunity for short-term investors to generate profits in financial markets has led to a significant increase in the number of traders in these markets in recent years. Therefore, searching for reliable methods to support traders in making decisions on the financial markets remains an active area of study in recent years (Deng *et al.*, 2021).

Forecasting in financial markets is a grand challenge due to the non-linear and dynamic nature of financial markets and the presence of numerous uncertainties that can contribute to the growth or decline of prices (Vaidya, 2020). Numerous variables, such as political and economic events and investor sentiment, have an impact on these uncertainties; however, the effect of the majority of these variables on price growth is unknown (Chang *et al.*, 2020). For this reason, accurate price forecasting in financial markets is challenging and necessitates innovative and reliable approaches. Based on this motivation, this study proposes novel hybrid neural network (NN)-based metaheuristics to control the enormous uncertainties in the financial and precious metals markets.

From the literature, current prediction models are incapable of meeting the requirements of today's complex financial markets. The results of some conventional approaches, such as the regression model for time series predictions, are not particularly accurate because it is always necessary to study other markets, such as the energy market, to identify ascending and descending patterns in the series charts by analyzing the direction of price movement (Xiao *et al.*, 2021). Technical analysis indicators, which are typically used to earn profits in the short term, are one of the approaches that are suitable for managing complex data with non-linear relationships and do not require specialized financial knowledge or the study of other markets (Chandar, 2022).

Indicators are auxiliary charts and mathematical functions that generate buy and sell signals where they extract data from price and time charts and input it into specialized mathematical functions; the result of this process signals allow analysts to make immediate decisions (Chandar, 2022). These signals are either buy or sell signals with a high error percentage. In other words, when the indicator indicates a buy, the price will decrease, and when it indicates a sell, the price will increase. Many false signals in this market, on the other hand, have significant negative effects on

Received: January 16, 2023. **Revised:** April 11, 2023. **Accepted:** April 12, 2023

industrial production and exacerbate volatility and uncertainty for producers and investors (Deng *et al.*, 2021).

With the advancement of science and technology and the use of artificial intelligence in the analysis of data obtained from technical analysis indicators, it is possible to play a significant role in resolving issues associated with predicting the price of financial markets and boosting the profitability of individuals, organizations, and countries (Gholizadeh *et al.*, 2022). In the field of artificial intelligence and machine learning, the use of deep learning (DL) methods is one of the most prevalent and widely employed techniques. DL is a highly effective technique that imitates the way the human mind learns certain topics focusing on artificial neural networks (ANNs) as the process of analyzing results and topics is faster and simpler (Gunnarsson *et al.*, 2021).

Although the literature is rich in the development of DL methods and ANNs, there are still weaknesses and disadvantages associated with these methods. Researchers have paid considerable attention to the fact that NN hyperparameters are determined using random search and grid search, which is one of the networks' weaknesses (Martia & Yasmine, 2021). These processes are time-consuming and necessitate more intricate mathematical calculations. Hence, this paper proposes a hybrid NN-based metaheuristic solution to this problem combining the strengths of different algorithms.

The proposed method is a combination of convolutional neural network (CNN) and bidirectional gated recurrent unit (BiGRU) which is abbreviated as CNN-BiGRU. In addition, the firefly algorithm (FA), which is one of the successful metaheuristic algorithms, is also used to enhance the performance of NNs while finding the optimal hyperparameters. Using this metaheuristic algorithm decreases the rate of computation while increasing accuracy. To examine the accuracy, precision, and validity of the proposed model, the method was compared to other single and hybrid DL and machine learning algorithms such as tree-based algorithms. The proposed model is evaluated using data extracted from the global gold market over 1 year. These data are derived from a 1-year analysis of signals from three widely used indicators among financial market analysts: moving average convergence divergence (MACD), Ichimoku, and moving average (MA). These indicators' signals were analyzed in 112 distinct modes, and the resulting data were recorded. The moth-flame optimization (MFO) algorithm, a successful metaheuristic algorithm, has been utilized for feature selection because the use of all variables in modeling can reduce the accuracy of the results.

In conclusion, the primary contribution of this study is the prediction of signals derived from technical analysis indicators in the global gold market using hybrid NN-based metaheuristics. When indicators produce false signals, investors may incur significant losses. Since this method improves signal accuracy, it can be extremely beneficial for gold market investors worldwide. In other words, investing in this market using this method can generate profits for users and serve as a decision-making tool. In actuality, this strategy can prevent investor losses and generate profits. Due to the algorithm's high precision, it can also be used to detect false signals that show investors the exact time to enter and exit the financial markets.

To accomplish the aforementioned objectives, the rest of this paper is organized as follows: In Section 2, the literature on the topic is explored. In Section 3, we examine the problem description and proposed our prediction model. Results and discussions are explained in Section 4. In Section 5, conclusions and management approaches are presented.

2. Literature Review

This study contributes to the body of knowledge in multiple ways. We first review the related studies to technical analysis and popular indicators in technical analysis. Then, we focus on DL and predictions on this subject. Finally, we study the role of metaheuristics in financial markets while reviewing the relevant studies.

2.1. Technical analysis and indicators

For many investors, technical analysis is a popular method for predicting stock prices and previous studies have shown that investors are more interested in using technical analysis than fundamental analysis. From the viewpoint of technical analysts, the historical performance of stock markets reflects future performance, and using such methods can make more profit (Chang *et al.*, 2020). Analysts adopt indicators to predict future stock prices by analyzing historical data (Deng *et al.*, 2021). However, Batten *et al.* (2018) found that the use and combination of a set of parameters significantly increase the ability to predict the price in some markets such as the gold market; however, they do not perform accurately in other financial markets such as the silver market. A popular approach showing high performance in the short term is technical indicators which are mathematical calculations based on historical price, volume, or other trading information. The purpose of these indicators is to predict the movement of the value of financial markets (Murphy, 1999).

MACD is one of the popular tools in the technical analysis (Khatua, 2016). This indicator is employed to aid in technical analysis. In addition, it serves as a method for identifying trends and signals in stock trading (Martia & Yasmine, 2021). Vaidya (2020) used the MACD to analyze the Nepal stock market. Agudelo Aguirre *et al.* (2021) used genetic algorithm (GA) to optimize some traditional indicators of technical analysis such as MACD in order to make more accurate predictions in stock markets by increasing the accuracy of the models. Chong *et al.* (2014) found that the MACD instrument can achieve significant positive returns when it uses optimized parameter values. In 2014, researchers evaluated the profitability of the MACD and the relative strength index (RSI). The use of MACD and RSI indicators is customary in the literature. Examining the profitability of two popular technical trading rules MACD and RSI in the Australian stock market, Nor and Wickremasinghe (2017) found that the MACD is generally weak in the Australian market, and the RSI is more accurate in Australian market and can be more profitable.

Another indicator in technical analysis is the MA, which is very popular among analysts. Research conducted in 2019 partially justifies the prevalent use of MA rules in the stock market (Kouaissah *et al.*, 2020). Sulistiawan and Rudiawarni (2020) examined the performance of the MA as a popular indicator in Indonesian stock markets. In 2021, researchers from China examined the role of the MA in the volume of asset pricing transactions. They find that the gap between the short- and the long-term MAs in terms of volume is returning strongly in the Chinese stock market (Ma *et al.*, 2021). Last but not least in technical analysis is the Ichimoku. In 2020, researchers examined the performance of the Ichimoku in the US capital market (Gurrib *et al.*, 2020). They used the Ichimoku cloud function as a trading model and compared the results with a simple buy-and-hold strategy.

2.2. DL methods

DL is an approach that is highly applicable in diverse fields of engineering (Gunnarsson *et al.*, 2021). One of its applications is in the classification of signals. The signals in each area are beneficial information that is very difficult to find. Zhou *et al.* (2021) classified electromagnetic signals for aircraft communications. Yan *et al.* (2022) classified emotions using physiological signals. Li *et al.* (2022) classified stock markets to identify the similarity between international stock markets.

In fundamental analysis and time series, the exchange rate is one of the most popular variables used as model input. Zhong and Enke (2017) expanded their model using prices between the US Dollar and four other currencies. Other researchers have used the price of digital currencies as an input variable (Dingli & Fournier, 2017). They used CNNs to predict the price direction of the next period according to the current price and were able to achieve 65% accuracy to predict the price in the next month. Other researchers have used energy market fluctuations, such as the oil market, to predict other financial markets, such as stock and price forecasts (Gillaizeau *et al.*, 2019). Using time and frequency domain mechanisms, they identified the "givers and receivers" of Bitcoin price volatility and discussed international diversification strategies in this context. Other researchers used long short-term memory (LSTM) networks perspective to predict stock prices. They also used the interest of investors to improve forecasting accuracy, e.g., price, volume, and indices as variables (Zhang *et al.*, 2021). LSTM is one of the techniques for learning sequences. They are usually less applicable to financial time series forecasting but are inherently appropriate for the field, and many researchers used this approach. Liu and Long (2020) used outlier robust extreme learning machine and LSTM to predict stock prices. Fischer and Krauss (2018) used LSTM to predict the orientation of a stock sample. Moskowitz *et al.* (2012) introduced an instantaneous time series momentum. Meher *et al.* (2021) used auto-regressive integrated moving average to forecast the stock prices of some selected pharmaceutical companies in India. Xiao *et al.* (2021) forecasted the Chinese stock market volatility development using six heterogeneous auto-regressive (HAR) models. Other researchers use HAR to forecast realized volatility in the Japanese spot stock markets (Maki & Ota, 2021).

Recently, the DL methods are highly active to study the spread of COVID-19 in literature (Moosavi *et al.*, 2022). In 2022, a researcher from Canada focuses on the impact of COVID-19 on Canada's stock market volatility (Xu, 2022). Cheng *et al.* (2022) used the Diebold–Yilmaz volatility network to analyze the volatility spillover index to show the effects of the COVID-19 pandemic. Díaz *et al.* (2022) investigated the impact of the Corona pandemic on the news of stock market fluctuations around the world. They found that the spread of the disease has a positive and significant effect on the volatility of the stock market.

2.3. Metaheuristic algorithms in financial markets

Effective forecasting of stock market prices and trends is an important issue in financial research for investors and shareholders who want to increase their investment returns. Therefore, optimizing and forecasting approaches to increase the accuracy of algorithms can play a significant role in improving the performance of financial markets (Islam *et al.*, 2021). For example, researchers presented a mixed support vector regression model using multi-output least squares optimized with sliding window, the hyperparameters of the proposed model using teaching-learning-based optimization, search for symbiotic organisms, and forensics-based investigations are optimized (Chou *et al.*, 2022). Tripathi *et al.* (2023) used an NN model to predict the stock market while achieving a more accurate prediction. Hence, the weight of the NN is optimally through the combination algorithm of the whale optimization algorithm and FA.

With the start of the corona virus epidemic and the economic crisis, other researchers predicted the companies that are likely to fail (Elhoseny *et al.*, 2022). For this purpose, they proposed an outlier detection model using a political political optimizer-based deep neural network (OD-PODNN). OD-PODNN aims to determine the financial status of a firm or company by involving several processes, namely preprocessing, outlier detection, classification, and metaparameter optimization, which uses isolation forest (iForest) to detect outliers.

The ability of NN in the field of prediction and the value of prediction in all fields have attracted researchers to increase the accuracy of predictions using NNs, especially in financial markets. For this purpose, Ghasemiyeh (2017) optimized the performance of NNs in stock price prediction using cuckoo search, GA, particle swarm optimizer (PSO), and bird mating optimizer algorithms. In another study, GA, PSO, and artificial bee colony (ABC) were used this time by Chandar (2021) to optimize the hyperparameters of ANNs in stock market forecasting.

In conclusion, many studies attempted to predict the process of changes in financial markets. However, the approaches used are mostly limited to time series algorithms, single DL algorithms, and technical analysis indicators, which can model past data. However, they did not show accurate predictions of future data. In this regard, this study suggests predicting the trend of changes in signals obtained from technical analysis indicators. The datasets obtained from technical analysis indicators are analyzed using various metaheuristic algorithms. Then, we optimize the hyperparameters of our DL model with metaheuristic algorithms. However, no forecasting/prediction studies develop our NN-based metaheuristic algorithms in this research area. As such, no research uses the signals obtained from technical analysis to predict the finance and precious metals markets as a research gap.

To fill it, this study represents hybrid NN-based metaheuristic methods in predicting the signals obtained from technical analysis indicators is the major contribution of this article. Using this approach and this model can act as a decision support tool for investors in the metal markets. Managers and investors can use this model to make large profits from capital markets and precious metals markets in a short time.

3. Problem Description

In financial markets, price forecasting is always a topic of interest for academics and professionals, which has led to the creation of several different methods to improve forecasting. Several approaches are available in the literature to identify future price movements, such as time series regressions. However, these approaches do not work well in reality due to the great uncertainty that exists when it comes to predicting the financial markets. Technical analysis is based on indicators, which are among the most commonly used approaches by analysts. Indicators extract information from price and time charts that are available to all financial market participants, then put them into special mathematical functions, and the output of this process signals for buying or selling. In a short time, these signals can yield huge profits for investors. However, these signals have a high probability of error, which means that the price will decrease after the indicator gives us a buy signal, or after the indicator gives us a sell signal, the price will decrease. Investors suffer huge losses due to the unreliability of these indicators. The article proposes a novel hybrid approach to predicting the financial markets and increasing

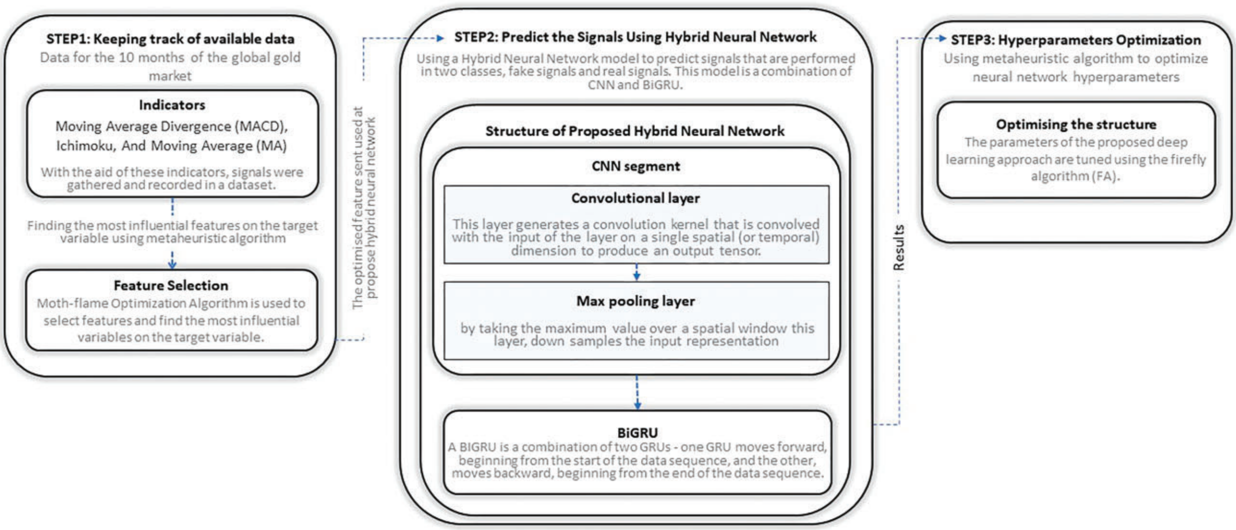

Figure 1: Structure of the proposed approach.

investors' profits instead of using fundamental analysis that requires specialized knowledge, time series regressions that require the study of other markets like the oil market, or indicators that are highly likely to be inaccurate.

Using an integrated DL approach, the proposed approach improves the performance of technical analysis indicators. Based on the obtained results, the gold market will be examined for 10 months (the first 10 months of 2020). It is either the result of an indicator that has been analyzed in different ways or the combination of different analyses of indicators. A total of 2638 records were obtained. We receive different signals from each of these indicators. As a result, our features and dataset can be analyzed in different ways. A total of 112 features were extracted from these indicators. These features were used to classify signals and predict fake signals from valid signals. Our target variable was stored in binary format, along with some of the other data. Our target variable takes the number 1 and is called a successful signal when the price has a lot of fluctuations and the signals that occur have moved up or down more than 20 pips. Otherwise, it takes the number 0 and is called a fake signal. All of these variables are recorded in an integer that is both binary and non-binary.

3.1. Proposed integrated model

Signals obtained from the indicators can be classified into two classes, fake signals, and true signals. This model was developed based on three popular indicators among financial market analysts, namely MACD, Ichimoku, and MA over 1 year in the global gold market. The signals obtained from these indicators were recorded in a dataset. It contains 112 features derived from different analyses and the combination of these three tools. For feature selection, the MFO algorithm is employed since the use of all these features can be impractical and difficult. Following this, we present NN-based metaheuristic algorithms as our prediction model which combines CNN and BiGRUs. The combination of these algorithms has also been expanded to improve the performance of CNN and BiGRU networks. To enhance the performance of the gated recurrent unit (GRU) network, the BiGRU network has been expanded, so CNN-BiGRU was used as a combination of these two networks. The CNN-BiGRU network processes the data after the MFO algorithm selects features. The CNN layer extracts the implicit information from short utterances and transmits the depth properties to the BiGRU layer. BiGRU models long-term correlation by modeling the dependency between features. The extracted global background property information vectors are fed to the output layer. It can also turn properties into a space that makes it easier to classify the output. By selecting the network hyperparameters optimally, the models perform better and have lower errors. Searching and selecting these hyperparameters manually may be extremely time-consuming, but it is not guaranteed that they will be optimal. Finding the optimal value for hyperparameters can be simplified using the FA. Figure 1 illustrates the structure of the proposed approach. The following sections discuss the different components of the proposed approach.

3.2. Analysis of indicators

As mentioned earlier, our methodology uses three indicators including MACD, Ichimoku, and MA which are explained as follows:

3.2.1. *Moving average convergence divergence*

MACD is a simple and effective indicator that helps traders to identify key price points and identify trends. Subtracting the 26-period MA from the 12-period MA is calculated by the MACD (Agudelo Aguirre *et al.*, 2021). An exponential moving average (EMA) creates more weight to new points than a MA. There are several ways to interpret MACD. Intersections, divergences, and rapid price increases/decreases are the most common interpretation methods. The MACD fluctuates above and below the zero line with convergence, cross-over, and divergence of MAs. When the MACD crosses its signal line and goes up or down, it produces technical signals. Signal lines, intersections of centerlines, and divergences can be used by traders to generate signals. This means that traders usually make securities trading when the MACD crosses its signal line and goes up and sells (or shortens) the securities when the MACD crosses the signal line and goes down. Equation (1) calculates MACD.

$$MACD_i\,(n) = \sum_{i=1}^{n} EMA_{k}\,(i) - \sum_{i=1}^{n} EMA_{d}\,(i) \tag{1}$$

In Equation (1), k and d show 12 and 26 EMA which are constructed based on historical quote data of the analyzed financial asset. In other words, the MACD line is created from the difference between EMA 12 and 26. Also, EMA is shown in Equation (2).

$$\mathrm{EMA}_n(i) = \frac{2}{1+n} \times p(i) + \left(1 - \frac{2}{1+n}\right) \times \mathrm{EMA}_n(i-1) \tag{2}$$

In Equation (2), n shows the number of data considered for the calculation of EMA and the price of the asset is shown by $p(i)$ on i^{th} day.

3.2.2. *Moving average*

MAs are also commonly employed in technical analysis. This strategy is utilized by financial market investors. When the short-term MA crosses the long-term MA, short-term prices will be relatively higher than long-term prices, and prices will rise. A rising intersection occurs when the shorter MA is less than the longer MA. When the shorter MA crosses the longer MA, a bearish cross occurs. A bullish cross-over indicates that stock prices will rise in the coming days, and the trader interprets this as a buy signal. A bearish intersection indicates a decline in stock prices over the next few days, and it is prudent to sell the security. Equation (3) shows a simple MA to reveal long-term trends and cycles over n time periods. In other words, this equation shows that the MA of a given day is the average price in the previous n days, and n represents the number of time periods. $p_n + p_{n-1} + \ldots + p_1$ also indicate the stock price on a previous n day.

$$\mathrm{MA}_n = \frac{(p_1 + p_2 + \ldots + p_n)}{n} \tag{3}$$

3.2.3. *Ichimoku*

In 1968, Goichi Hosoda developed Ichimoku strategy which translates to a balance chart at a glance. Ichimoku consists mainly of five lines and five elements (Gurrib *et al.*, 2020). These elements are Senkou-span B, Senkou-span A, Chikou-span, Kijun sen, and Tenkan-sen. By using historical near, high, and low prices, these five elements can be calculated using Kumo model as a cloud one (Deng *et al.*, 2021). With two different colors filling the gap between Senkou-span A and B openings, this cloud illustrates investors' different feelings. We can say that the market is bullish if the closing price is above the Ichimoku cloud at the moment. Ichimoku clouds indicate stabilization when the current closing price is within them. The market is in a downtrend if the closing price is below the Ichimoku cloud. There are many similarities between the Ichimoku elements and the standard MAs. Tenkan-sen and Kijun-sen are similar to the short- and medium-term mobile ages, respectively. Senkou-span A and B are similar to long-term MAs. The Chikou-span indicates whether a trend is likely to occur or not. Note that to move the Chikou and Senkou openings A and B openings, 26 cycles of backward or forward shifts are required. This means that the transfer will take place in 25 periods. Description of Ichimoku's five elements with parameter setting (9, 26, 52) are shown in Equations (4–8). $p_{\mathrm{H}}(t)$, $p_{\mathrm{L}}(t)$, and $p_{\mathrm{c}}(t)$ are the highest, lowest, and closing price/rate at time period t, respectively.

Equation (4) is about Tenkan-sen ($\mathcal{TK}$) to show the middle value of the highest high and lowest low for the past nine periods where the current period is included.

$$\mathcal{TK}(t) = \frac{\max\{p_{\mathrm{H}}(t)\} + \min\{p_{\mathrm{L}}(t)\}}{2}, \quad t - 8 \le \mathcal{T} \le t \tag{4}$$

Equation (5) is about Kijun-sen ($\mathcal{KJ}$) to show the middle value of the highest high and lowest low for the past 26 periods where the current period is included.

$$\mathcal{KJ}(t) = \frac{\max\{p_{\mathrm{H}}(t)\} + \min\{p_{\mathrm{L}}(t)\}}{2}, \quad t - 25 \le \mathcal{T} \le t \tag{5}$$

Equation (6) is about Chikou-span ($\mathcal{CK}$) to show the current closing price time-shifted backwards (into the past) for 26 periods where the current period is included.

$$\mathcal{CK}(t - 25) = p_{\mathrm{c}}(t) \tag{6}$$

Equation (7) is about Senkou-span A ($\mathcal{SKA}$) to study the middle value of the current Tenkan-sen and Kijun-sen, and it is then time-shifted forwards (into the future) for 26 periods where the current period is included.

$$\mathcal{SKA}(t) = \frac{\mathcal{TK}(t-25) + \mathcal{KJ}(t-25)}{2} \tag{7}$$

Equation (8) is about Senkou-span B ($\mathcal{SKB}$) to evaluate the middle value of the highest high and lowest low value for the past 52 and future 26 periods where the current period is included.

$$\mathcal{SKB}(t) = \frac{\max\{p_{\mathrm{H}}(t)\} + \min\{p_{\mathrm{L}}(t)\}}{2}, \quad t - 76 \le \mathrm{T} \le t - 25 \tag{8}$$

3.3. Feature selection by MFO metaheuristic algorithm

MFO algorithm is a powerful metaheuristic algorithm for feature selection that finds a solution to a problem based on the behavior of moths around a flame (Mirjalili, 2015). Moths are unique in their ability to navigate at night. They evolved to use moonlight for night-time flight. They navigate using a mechanism called transverse orientation. Using this technique, a moth flies at a constant angle relative to the moon. Despite their transverse orientation, butterflies typically fly in a fatal spiral path around artificial lights. In this model, moths are the actual search agents that traverse the search space, and flames are the highest ever moth position (Bhadoria & Marwaha, 2020; Zhang *et al.*, 2022). Following is the pseudo-code of the Algorithm 1.

Algorithm 1: Moth-flame optimisation algorithm (MFO)

```
The initialisation of solutions and the required parameters of the algorithm.
Set it = 1;
Evaluate the fitness of solutions;
while iteration it ≤ it_max
    if it == 1 then
        F ← sorted(M);
        OF ← sorted(OM);
    else
        F ← sorted(F_{t-1}, M_t);
        OF ← sorted(OF_{t-1}, OM_t);
    end
    for i-th
        Determine D for the corresponding Moth and Flame;
        Update Moth (i);
    end
    it = it + 1
end
F ← sorted (F_{t-1}, M_t);
Return the most optimal solution;
```

3.4. Proposed NN-based metaheuristics

3.4.1. *Convolutional neural network*

CNN can be characterized as leading NNs with deep structures (Houssein *et al.*, 2022). There are five layers in a CNN, which are an input layer, a convolution layer, a pooling layer, a fully connected

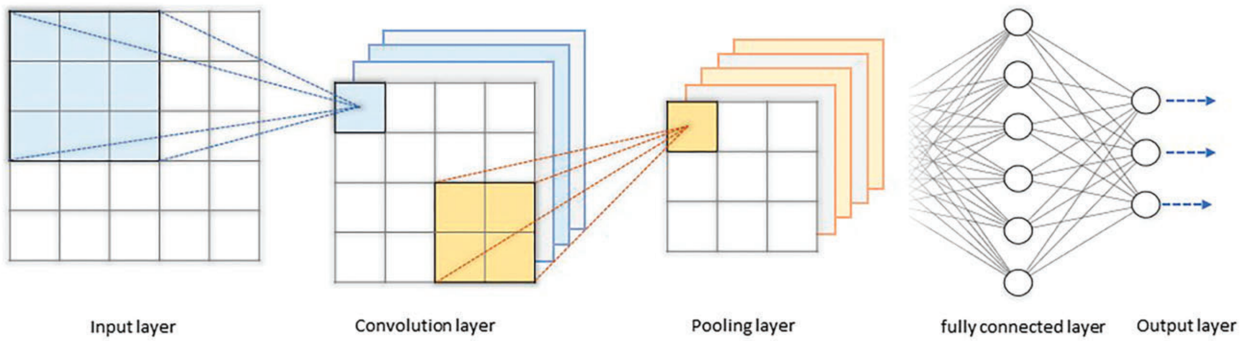

Figure 2: Structure of CNN.

layer, and an output layer. Outputs are generated after receiving the properties in the output layer. The CNN structure is shown in Equation (9) is related to the calculation formula of CNN. $\mathcal{N}$ indicates the output size. Input size is shown by $\mathcal{W}$, convolution kernel size is shown by $\mathcal{F}$, padding value size is shown by $\mathcal{P}$, and step size is shown by $\mathcal{S}$.

$$\mathcal{N} = (\mathcal{W} - \mathcal{F} - 2\mathcal{P})/\mathcal{S} + 1 \tag{9}$$

From the input data, features are extracted by the convolution layer (Liu *et al.*, 2022). Generally, convolution layers consist of convolution kernels, convolutional layer parameters, and activation functions. Convolution is the most significant and unique layer on CNN. By using convolution kernels, the convolution layer can extract features from input variables. Essentially, a convolution kernel extracts properties. Convolution kernels have a smaller scale than input matrices. Convolution layer instead of general matrix operation produces the feature map using convergence operations. Equation (10) shows the calculation of each element in the feature map. In Equation (10), $x^{\text{out}}_{i,j}$ is the output value in row i and column j of the feature map. The value in row i and column j of the input matrix is shown as $x^{\text{in}}_{i+m,j+n}$. The active function is selected by $f_{\text{cov}}(0)$. $w_{m,n}$ shows the weight in row m and column n for the convolution kernel. Also, the bias of the convolution kernel is shown by b.

$$x^{\text{out}}_{i,j} = f_{\text{cov}}\left(\sum_{m=0}^{k}\sum_{n=0}^{k} w_{m,n}x^{\text{in}}_{i+m,j+n} + b\right) \tag{10}$$

The input matrix uses multiple kernels for the convolution layer. Each convolution kernel extracts a feature from the input matrix to create the feature map. The pooling layer then reduces the length and width of the previous feature map. With downsampling, the computational efficiency is also improved. Through the pooling layer, the convolutional layer can reduce the output vectors of features (Mamoudan *et al.*, 2022). It is also possible to improve the results. As CNN is good at extracting grid data features, *m* variables of any type were expanded to *n* stations in order to obtain a matrix of *m* rows and *n* columns. Figure 2 illustrates the overall structure of the CNN.

3.4.2. *Gated recurrent unit*

A GRU is one of the algorithms developed for solving the short-term memory problem of recurrent neural networks. There are internal mechanisms within the GRU network known as gate (Liu & Long, 2020). Gates control the flow of information and determine what sequential data should be retained and what data should be deleted. Thus, important information is passed along the sequence chain to achieve the desired result. An important component of the GRU network structure is the cell state, which is considered the internal memory of the network. Information is transmitted over the network by the cell state, which is updated by structures known as gates. There are two gates in the GRU network where the gates are referred to as the reset gate and the update gate. Data are transmitted over the GRU network using a hidden state where this network also uses the sigmoid and tan*h* functions.

Update gates determine how much past information, i.e., the information we had in the previous steps, will be added to the network. In this gate, the value of the new input (x_t) together with the latent state value of the previous step (h_{t-1}) is multiplied by its corresponding weight and then added together and entered into a sigmoid function so that the output is in the range [0,1] (Niu *et al.*, 2022). During network training, these weights are updated each time so that only useful information is added to the network. As a final step, the gate output is updated with the latent state of the previous step pointwise multiplication, which is used to calculate the gate output later on.

Reset gate decides how much information from the previous step, i.e., the information from previous steps, is forgotten. Here, too, the new input value (x_t), together with the cache value of the previous step (h_{t-1}), is multiplied by its corresponding weight and then added together and entered into a *sigmoid* function so that the output is in the range [0, 1]. To take the difference with the upgrade gate is that the weights at which the input value and the latent state of the previous step are multiplied are different, which means that the output vectors here will be different from the output vectors we have in the upgrade gate. The output of the reset gate is then pointwise multiplied by the latent state of the previous step. Finally, this output vector enters a *tanh* function.

Finally, the output gate output vector is used again to obtain a new hidden state. In this step, the element-wise inverse version of the update gate output is pointwise multiplied by the output obtained from the *tanh* function (r). The purpose of the update gate is to detect what new information obtained in this step is stored in the new cache. Finally, the output of this step is added to the output of the update gate multiplication point with the previous latent state (u) and creates a new latent state. This stealth mode can be used as the final output (last step). All the above information is shown in Equations (11–14) where W_z, W_r, and W show learnable weight matrices. The previous hidden state is shown as h_{t-1}. x_t is shown the input vector. σ and *tanh* are the *sigmoid* and *tanh* activation function in these equations. Finally, b_z, b_r, and b are biases. Figure 3 shows the structure of this method.

$$z_t = \sigma\ (W_z.[h_{t-1}, x_t] + b_z) \tag{11}$$

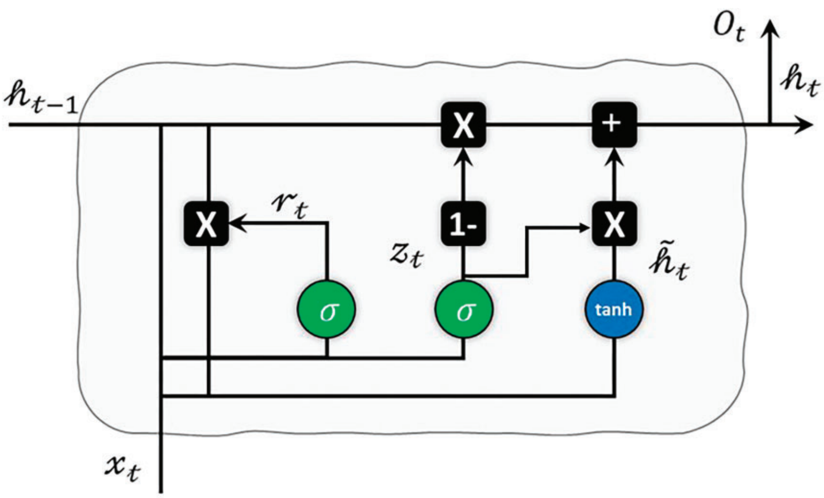

Figure 3: GRU structure.

$$r_t = \sigma\ (W_r.[h_{t-1}, x_t] + b_r) \tag{12}$$

$$\widetilde{h}_t = tanh\ (W.[r_t * h_{t-1}, \mathrm{x_t}] + b) \tag{13}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h}_t \tag{14}$$

3.4.3. *Bidirectional gated recurrent unit*

Among the main problems associated with the GRU network which led to the development of algorithms for improving its performance is that the GRU network takes into account only the input series in one direction, which makes it unsuitable for modeling the display of features comprehensively. As a result, another network called the BiGRU was developed to address this breach in the GRU network. To improve the performance of the GRU network, the BiGRU structure has been developed by modeling the input series from the forward directions and backward directions. Equations (15) and (16) show the forward and backward structures in the BiGRU network, which are represented by $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$, respectively (Meng *et al.*, 2021). The combination of the forward structure and the backward structure together produces y_t, which is shown in Equation (17). Figure 4 shows BiGRU's structure.

$$\overrightarrow{h}_t = \overrightarrow{GRU}\,(x_t, h_{t-1}) \tag{15}$$

$$\overleftarrow{h}_t = \overleftarrow{GRU}\,(x_t, h_{t-1}) \tag{16}$$

$$y_t = \left[\overrightarrow{h}_t, \overleftarrow{h}_t\right] \tag{17}$$

3.5. Optimization tools

One of the main weaknesses of NNs is stopping at local optimal points. In general, finding optimal points in DL algorithms and NNs is done in the form of random search, pattern search, and grid search methods. However, metaheuristic algorithms can find a global optimum solution using diversification and intensification phases. These approximate solutions found by metaheuristics are more powerful than random search, pattern search, and grid search methods. Metaheuristic algorithms intelligently reduce the search space leading to convergence to an optimal solution in a short time.

To improve the performance of the proposed model, an optimization strategy is used to determine the optimal model structure. This study uses the FA, one of the most efficient metaheuristic algorithms. Since its development, FA has attracted much attention and has been used in a wide range of applications. This algorithm is based on a physical formula of decreasing light intensity with increasing distance squared (El-Shorbagy & El-Refaey, 2022). However, as the distance from the source of light increases, light absorption causes the light to diminish. These phenomena are related to the objective function that must be optimized. Consequently, the base FA can be expressed as shown in Algorithm 2.

The "Initialization" function initializes the firefly population. Randomization is typically used to perform this initialization. Within the while loop, the firefly search process consists of the following steps: Firstly, the "NewAlpha" function modifies the initial value of parameter α. In the FA, this step is optional. In the second step, the "Evaluation" function measures the quality of the solution. Thirdly, the "Order" function sorts the firefly population by fitness. In addition, the "BestFA" function identifies the best individual in the population. As a final step, the "MoveFA" function moves the firefly positions. There is a tendency for fireflies to move towards the most attractive individuals. A maximum number of iterations $it_{\max}$ controls the firefly search process.

3.6. Implementation of our prediction model

The implementation of the proposed approaches will be discussed in this section. Our research used Python open-source libraries. TensorFlow, an open-source software library offered by Google, and the Keras Toolkit were used to develop the models. Sklearn, NumPy, Pandas, and Matplotlib were additionally utilized to process, manipulate, and visualize data. A Core I7 11800H CPU, 16 GB of RAM, and an RTX 3060 graphics card were used for this study.

3.6.1. *Experiment and evaluation*

A cross-validation approach is performed in this study. To evaluate the accuracy and validity of the proposed model, precision, acc, recall and F1-score were used. These tests are based on the area under the curve, accuracy, true positive rate (T_p), false positive rate (F_p), true negative (T_n), false negative (F_n), accuracy, recall and f-measure. Accuracy is the most important metric, and it is the ratio between the total number of correctly predicted observations to the total number of observations. Precision, Accuracy, Recall, and F1 score calculate as Equations 18 to 21 as follows:

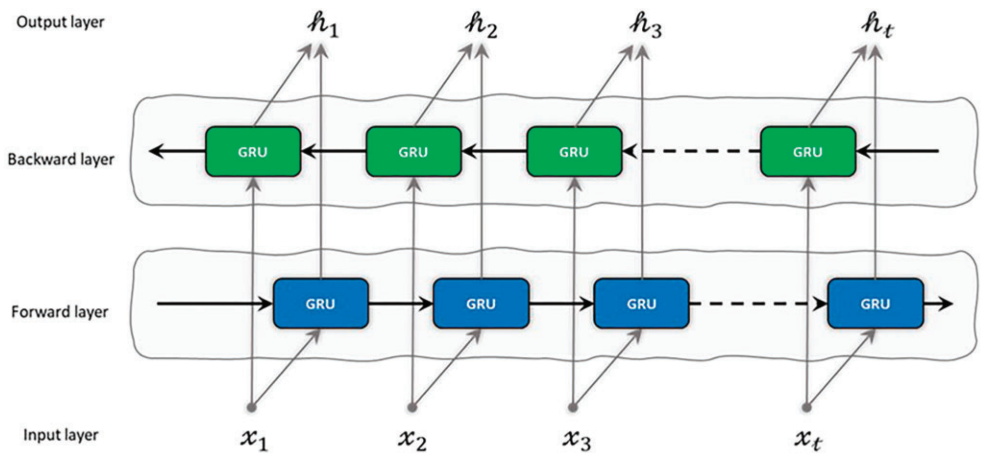

Figure 4: BiGRU structure.

Algorithm 2: Firefly Algorithm (FA)

The initialisation of solutions and the required parameters of the algorithm.

Set $it = 1, s^*, \gamma = 1.0$ ◃ it *iteration counter,* s^* *is the best solution and* γ *is attractiveness*

$\mathbb{P} \longleftarrow Initialization(\,)$; ◃ *generate initial solutions*

Evaluate the fitness of solutions;

while *iteration* $it \leq it_{max}$

  $\alpha^{it} = NewAlpha(\,)$; ◃ *Identify a new value for* α

  $Evaluation(\mathbb{P}^{it}, f(s))$; ◃ *Evaluate* s

  $Order(\mathbb{P}^{it}, f(s))$; ◃ *Sort* s

  $s^* = BestFA(\mathbb{P}^{it}, f(s))$; ◃ *Find the best solution*

  $\mathbb{P}^{it+1} = MoveFA(\mathbb{P}^{it})$; ◃ *Vary the attractiveness*

  $it = it + 1$

end

$$Accuracy = \frac{\mathcal{T}_p + \mathcal{T}_n}{\mathcal{T}_n + \mathcal{T}_p + \mathcal{F}_p + \mathcal{F}_n} \tag{18}$$

$$Precision = \frac{\mathcal{T}_p}{\mathcal{T}_p + \mathcal{F}_p} \tag{19}$$

$$Recall = \frac{\mathcal{T}_p}{\mathcal{T}_p + \mathcal{F}_n} \tag{20}$$

$$\text{F1 score} = \frac{2\ \times precision \times recall}{precision + recall} \tag{21}$$

Also, in order to compare the performance of the proposed model in predicting financial markets, some other single algorithms of machine learning and DL have been used, which include XGBoost, random forest, decision tree, support vector machine (SVM), K nearest neighbor (KNN), multi-layer perceptron (MLP), and CNN.

(i) **XGBoost** is an algorithm that is recently used in the field of machine learning for use in regression and classification problems, which has very fast performance and very high accuracy. The XGBoost algorithm is an implementation of decision tree gradient boosting designed for high speed and efficiency (Chou *et al.*, 2023).
(ii) **Random forest** is a machine learning algorithm for classification and regression problems that often provides very good results even without adjusting its metaparameters. This algorithm, based on a structure consisting of many decision trees, works on the time of training and the output of classes (classification) or for the predictions of each tree separately (Ray *et al.*, 2023).
(iii) **SVM** is another supervised algorithm in machine learning that can be used for both classification and regression problems. In SVM, each data sample is plotted as a point in *n*-dimensional space on a data scatterplot (*n* is the number of features a data sample has) (Koo & Shin, 2018). The value of each data attribute specifies one of the coordinate components of the point on the graph. Then, by drawing a straight line, it categorizes different and distinct data from each other.
(iv) **KNN** uses feature similarity to predict the values of new data points. In other words, this algorithm assigns a value to the new points (new data) based on its correspondence with the points of the training set (Li *et al.*, 2022).
(v) **MLP** is one of the most basic neural models that simulate the transfer function of the human brain. In this type of NN, most of the network behavior of the human brain and the propagation of signals have been considered in it, and hence, they are sometimes called feedforward networks (Pourkhodabakhsh *et al.*, 2022). Each of the nerve cells of the human brain, known as neuron, after receiving the input, performs a process on it and transmits the result to another cell. This behavior continues until a certain result is achieved, which will probably eventually lead to a decision or processing.

3.6.2. *Data description*

We obtained the data from the analysis and combination of MACD, Ichimoku, and MA indicators. We have analyzed them by different indicators in different ways. The results are signals that make up the dataset used in this study. We extracted 112 features from these analyses. There is variation between data categories. We used these features to analyze global gold price fluctuations and recorded them for 10 months. After cleaning the data, we got a total of 2638 records.

The main purpose of this study was to use these features to classify signals and predict fake signals from valid signals to predict the future price of financial markets. Our target variable takes the number 1 to show that it is a successful signal when the price fluctuates a lot and the signals that occur move more than 20 pips up or down. Otherwise, it takes the number 0 and is called a fake signal.

Figure 5 shows the histogram of some of these variables. In this figure, x4 shows some information about Ichimoku. This variable indicates whether the Ichimoku cloud was bullish or bearish when the cross occurred. The number 1 indicates the bullishness and the number 0 indicates the bearishness. The variable x16 indicates whether the candle is bullish or

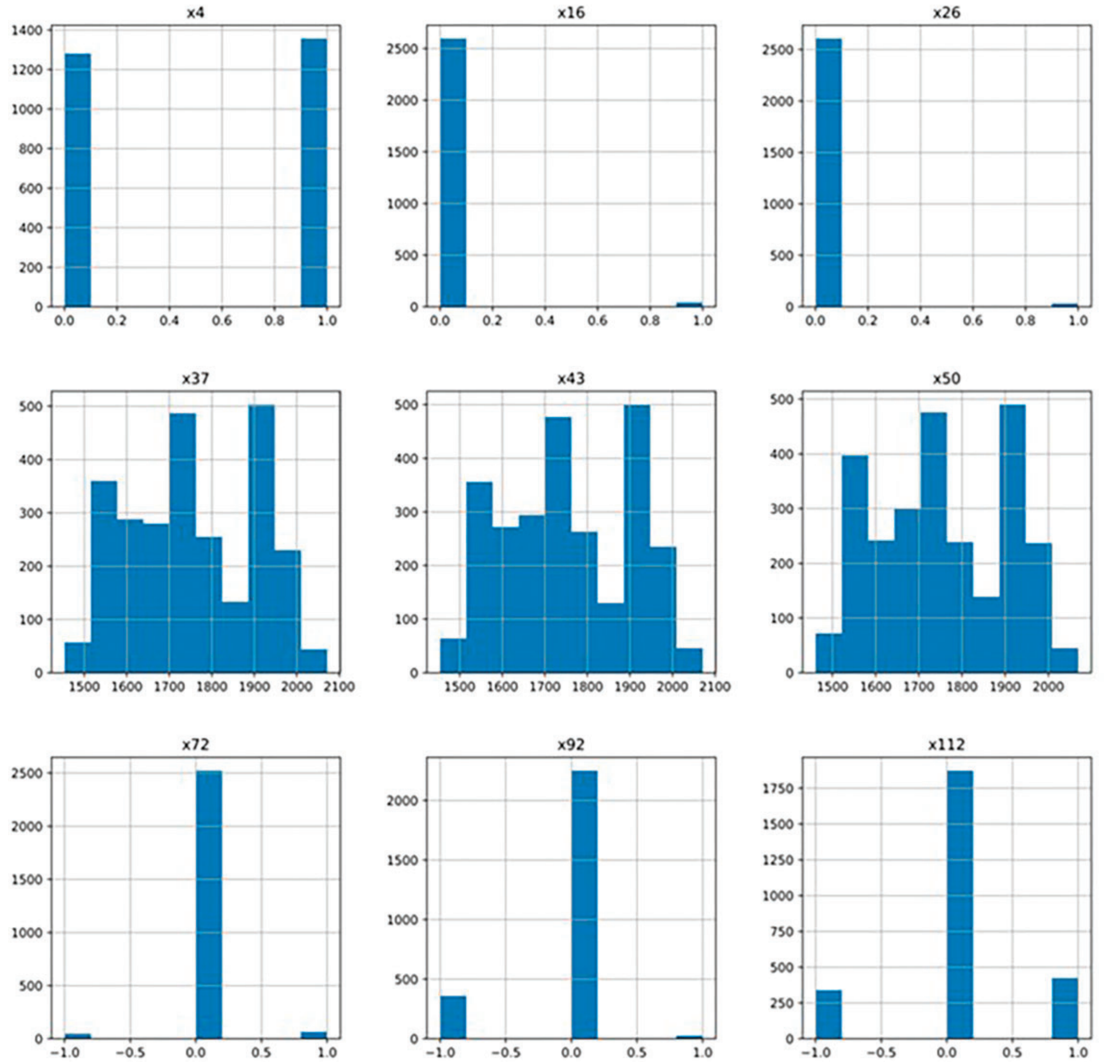

Figure 5: Data description.

bearish during the cross. X26 indicates whether the $\mathcal{TK}$ and $\mathcal{KJ}$ were equal to each other before the crossing, if they were equal to each other, it would be 1, otherwise it would be 0.

X37 represents the pivot price. Pivot points are one of the indicators of technical analysis or calculations that traders use to determine the general trend of the market in different time frames. This index shows the average maximum and minimum price and the closing price of the previous trading day. X43 corresponds to the EMA 9, which calculates the average price over the previous nine candles. X50 shows a pivot to the previous 47 candles.

X72 tells us the lowest or highest price has occurred in the previous nine candles. If we had the highest price, it would be 1, if we had the lowest price, it would be −1, and otherwise it would be 0. X92 shows the thinning process. When cross-over occurs, three states may occur. It is either an ascending line, which assigns the number 1, or a descending line, which assigns the number −1, and is 0 otherwise. X112 shows the location of $\mathcal{KJ}$ and cloud. If $\mathcal{KJ}$ is above the cloud, the number will be 1, if it is on one of the cloud lines, the number will be 0, and if it is below the cloud, it will be −1.

3.6.3. *Data preprocessing*

Performing machine learning (ML) projects require data preprocessing, including data cleansing, training, testing split, normalization, feature selection, etc. As a first step, it was checked whether there were any missed values in the data, and it was found that there were none in the data used. It is then decided that 80% of the data will be used for training, while 20% will be used for testing. In addition, since machine learning algorithms are sensitive to different feature scales, we normalized the data. Next, we selected the feature using the MFO algorithm, which is part of the proposed method. To demonstrate the superiority of the proposed model in the feature selection (which is done by MFO), we performed different algorithms using its optimal feature set for modeling and compared the results.

3.6.4. *Trained ML methods*

Our proposed model was compared with XGBoost, random forest, decision tree, SVM, KNN, MLP, and CNN. The presented approach is the hybrid of CNN and BiGRU, which is trained in each of the feature sets selected by the MFO, PSO, GA, and ANOVA *F*-test.

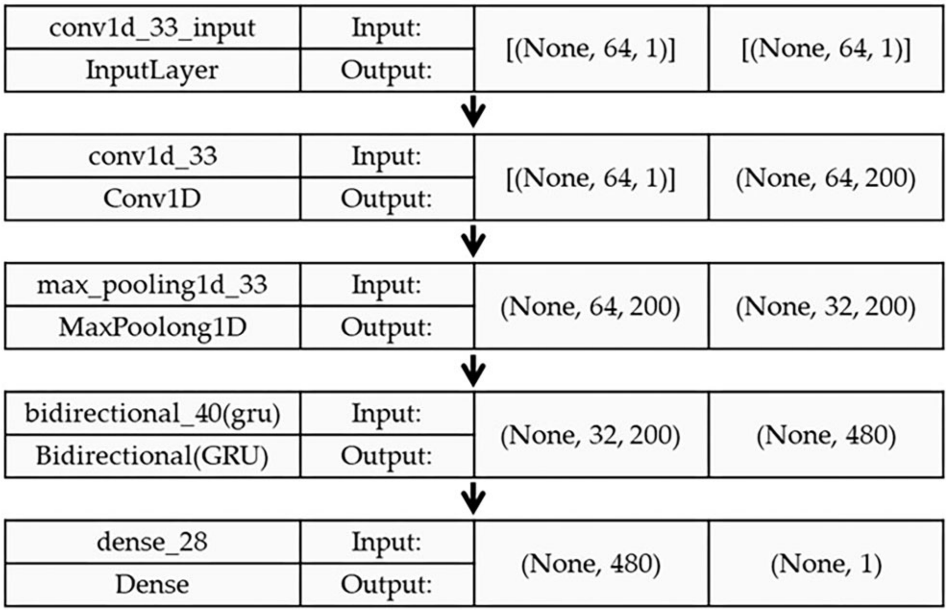

Figure 6: The network structure of CNN_BiLSTM.

3.6.5. *Network structure*

As mentioned previously, the suggested method for this article is to optimize the CNN and bidirectional long short term memory (BiLSTM) network with the FA and MFO algorithm on selected features. Figure 6 depicts the structure of the CNN BiGRU network utilized in this paper. The initial layer is composed of a conv1D (one-dimensional convolutional layer). In the subsequent layer, the output of the convolutional layer is passed to a MaxPooling1D layer. The pooling output enters the BiGRU layer in the third layer, and at the conclusion of the output, this layer enters the output layer.

3.6.6. *Hyperparameter optimization*

In this paper, the FA is used to determine the hyperparameters of the network. The hyperoptimized parameters include the type of activation function in the GRU layers, the number of neurons, the number of filters, and the type of activation function in the convolutional layer. In terms of the number of neurons and filters, the search space is [40,80,100,120,140,160,180,200,220,240,260] and in terms of the activation function, it is [relu, selu, tan*h*, sigmoid, linear]. It was selected that the number of GRU neurons would be 240, the number of convolutional filters would be 200, and both layers would be activated by tan*h*. Table 1 lists all the hyperparameters used in the network structure.

Table 1: Hyperparameters used by this study.

Hyperparameters	Value
Convolution layer filters	200
Convolution layer kernel_size	2
Convolution layer activation function	tan*h*
Pooling layer pool_size	2
Number of hidden units in BiGRU layer	240
BiGRU layer activation function	tan*h*
Number of hidden units in the output layer	1
Output layer activation function	sigmoid
Batch_size	32
Learning rate	0.001
Optimizer	Adam
Loss function	Binary cross entropy
Epochs	100

4. Result and Discussion

For the purpose of confirming the performance of our model (CNN-BiGRU-FA) in predicting the signals, several DL and machine learning algorithms were tested. Trial and error is used to optimize these algorithms' hyperparameters, so they have the lowest potential error. To evaluate the performance of the proposed model, we compared its performance with other DL and machine learning algorithms such as XGBoost, random forest, decision tree, SVM, KNN, MLP, and CNN. Evaluation criteria such as precision, accuracy, recall, F1-score and ROC-AUC have been used to evaluate the performance of these algorithms. The higher the value of these evaluation criteria, the less error and the better the model performance. In other words, these criteria show the model fit, and the closer they are to 1, the better the fit and the better the model performance.

Since feature selection is one of the important issues in training machine learning models, this paper proposes an MFO algorithm to select variables that have a reasonable and optimal dependence on the response variable. Table 2 shows the performance of DL and machine learning models for predicting signals from technical analysis indicators using the MFO algorithm. As the Table 2 shows, in selecting the features of the MFO, CNN-BiGRU has the best result with an accuracy of 93%. Also, the results show that F1-score and ROC-AUC, and the CNN-BiGRU have the performance with the rate of 0.90, 0.84, 0.87, and 0.90.

Figure 7 shows the graph of the results of feature selection with MFO algorithm. As it is known, the proposed model shows better results in terms of accuracy, F1-score, and ROC-AUC.

To confirm the effectiveness of the MFO in feature selection, DL and machine learning models have been investigated with PSO and GA thorough ANOVA *F*-test algorithm. Tables 3–5, generally highlight the performance of the proposed method in comparison with GA, PSO, and ANOVA *F*-test. In this regard, the *F*-value of

Table 2: Results of proposed model with MFO algorithm.

	Feature selection with MFO algorithm				
	Accuracy	Precision	Recall	F1-score	ROC-AUC
SVM	**0.81**	**0.59**	**0.88**	**0.71**	**0.83**
CNN-BiGRU	0.93	0.88	0.85	0.87	0.91
CNN	**0.90**	**0.76**	**0.91**	**0.83**	**0.91**
Random forest	0.90	0.75	0.89	0.81	0.89
KNN	**0.81**	**0.65**	**0.57**	**0.61**	**0.73**
MLP	0.85	0.74	0.65	0.69	0.79
Decision tree	**0.85**	**0.74**	**0.66**	**0.70**	**0.79**
XGBoost	0.83	0.66	0.70	0.68	0.79

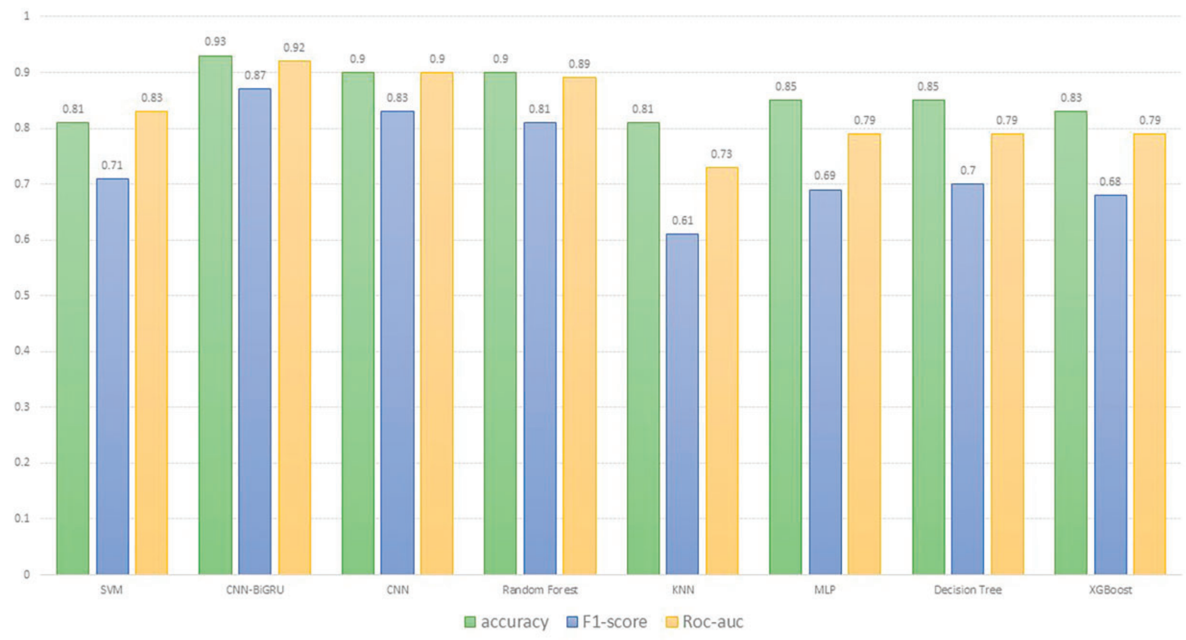

Figure 7: Results of CNN-BiGRU versus other algorithms with MFO algorithm.

Table 3: Results of proposed model with GA.

	Feature selection with GA				
	Accuracy	Precision	Recall	F1-score	ROC-AUC
SVM	**0.56**	**0.25**	**0.35**	**0.29**	**0.50**
CNN-BiGRU	0.92	0.80	0.84	0.86	0.90
CNN	**0.87**	**0.70**	**0.88**	**0.78**	**0.87**
Random forest	0.89	0.75	0.86	0.80	0.88
KNN	**0.72**	**0.37**	**0.11**	**0.17**	**0.52**
MLP	0.77	0.65	0.27	0.38	0.61
Decision tree	**0.85**	**0.70**	**0.76**	**0.73**	**0.82**
XGBoost	0.82	0.66	0.59	0.62	0.74

Table 4: Results of proposed model with ANOVA *F*-test.

	Feature selection with ANOVA *F*-test				
	Accuracy	Precision	Recall	F1-score	ROC-AUC
SVM	**0.56**	**0.25**	**0.35**	**0.29**	**0.50**
CNN-BiGRU	0.93	0.83	0.84	0.83	0.90
CNN	**0.87**	**0.69**	**0.88**	**0.77**	**0.87**
Random forest	0.89	0.75	0.87	0.80	0.88
KNN	**0.72**	**0.37**	**0.11**	**0.17**	**0.52**
MLP	0.80	0.66	0.43	0.52	0.68
Decision tree	**0.85**	**0.71**	**0.72**	**0.71**	**0.81**
XGBoost	0.82	0.64	0.71	0.67	0.79

Table 5: Results of proposed model with the PSO.

	Feature selection with PSO				
	Accuracy	Precision	Recall	F1-score	ROC-AUC
SVM	**0.72**	**0.27**	**0.33**	**0.29**	**0.68**
CNN-BiGRU	0.90	0.86	0.84	0.84	0.90
CNN	**0.90**	**0.64**	**0.83**	**0.72**	**0.87**
Random forest	0.88	0.79	0.91	0.84	0.90
KNN	**0.60**	**0.56**	**0.64**	**0.59**	**0.52**
MLP	0.84	0.68	0.65	0.66	0.69
Decision tree	**0.82**	**0.77**	**0.82**	**0.79**	**0.82**
XGBoost	0.91	0.80	0.88	0.83	0.79

Table 6: Results of optimized CNN-BiGRU with metaheuristic algorithms.

	Optimized CNN-BiGRU				
	Accuracy	Precision	Recall	F1-score	ROC-AUC
CNN-BiGRU	0.93	0.88	0.85	0.87	0.91
CNN-BiGRU-FA	**0.96**	**0.91**	**0.88**	**0.89**	**0.94**
CNN-BiGRU-GWO	0.94	0.90	0.87	0.88	0.92
CNN-BiGRU-EWA	**0.95**	**0.85**	**0.90**	**0.87**	**0.91**
CNN-BiGRU-ABC	0.96	0.86	0.84	0.84	0.90

each attribute is calculated from the variance of the data and then converted to *P*-value, which determines the importance of an attribute. This method calculates the data variance from two paths and then examines whether these variances are different or not. If they are different, it means that the mean of the properties in the classes is significantly different. If they are close to each other, it shows that, the mean of the feature in different classes is not significantly different and cannot be a good feature to separate the data of two or more classes.

In Table 3, the accuracy of our proposed algorithm is 0.92, which is 0.01 less accurate than the MFO algorithm. Based on the precision criterion, MFO was better than GA. The results show that in *F*1-score, GA was able to show 0.86, which is 0.01 less than the MFO. If we use GA for feature selection, the recall and ROC-AUC would be 0.84 and 0.90, respectively, which have decreased by 0.01 compared to the MFO.

In Table 4, the accuracy of the proposed algorithm is 0.93, which is equal to the MFO algorithm. In the precision, ANOVA *F*-test was able to show 0.83 in the proposed algorithm, which was 0.05 less accurate than the MFO. The results show that in *F*1-score, ANOVA *F*-test was able to show an accuracy of 0.83, which is 0.04 less than the MFO. If we use ANOVA *F*-test for feature selection, the recall and ROC-AUC would be 0.84 and 0.90, respectively, which are reduced by 0.01 and 0.01 respectively compared to the MFO.

Table 5 shows the results of the models when using the PSO algorithm as the feature selection algorithm. In Table 5, the accuracy of our proposed algorithm is 0.90, which is 0.03 less accurate than the MFO algorithm. In the precision, PSO was able to show a 0.86 in the proposed algorithm, which was 0.02 less than the MFO algorithm. The results show that in *F*1-score, PSO was able to show 0.84, which is 0.03 less than the MFO algorithm. If we use PSO for feature selection, the recall and ROC-AUC would be 0.84 and 0.90, respectively, which have decreased by 0.01 in comparison with the MFO.

Figure 8 shows the accuracy of different models using the feature selection with MFO, GA, PSO, and ANOVA *F*-test. As shown in Fig. 8, CNN-BiGRU algorithm has the highest accuracy in all feature selections.

Figure 9 shows a trade-off between the results of CNN-BiGRU algorithm in terms of accuracy, precision, recall, *F*1-score, and ROC-AUC. As shown in Fig. 9, MFO algorithm shows the best performed in all criteria for the feature selection.

After reviewing the performance of machine learning and DL models with different feature selection algorithms, here in table 6, we review different algorithms to optimize the hyperparameters of the proposed model. The hyperparameters of CNN-BiGRU have been optimized using Gray Wolf Optimizer (GWO), Artificial Bee Colony Algorithm (ABC), Earthworm Optimization Algorithm (EWA), BAT algorithma (BAT) and Firefly Algorithm (FA), the results of which are shown in Table 6. The results show that FA is able to perform better than GWO with an accuracy of 0.96. Also, *F*1-score and ROC-AUC in the CNN-BiGRU-FA are equal to 0.91, 0.88, 0.89, and 0.94, respectively. From the results, GWO is able to make a more accurate prediction. The results show that CNN-BiGRU-FA has a better performance than CNN-BiGRU-EWA and CNN-BiGRU-ABC. The comparison between different algorithms confirms that CNN-BiGRU-FA is able to predict signals with great accuracy compared to other algorithms and models.

Figure 10 shows the results of the optimization algorithms. When the hyperparameters of CNN-BiGRU algorithm are optimized using FA, it shows better results in accuracy, precision, recall, *F*1-score, and ROC-AUC. This comparison is done by the GWO as shown in this figure.

5. Managerial Insights

Contrary to previous literature that used time series data in regression models to predict prices in financial markets, this study presents a new approach in predicting price fluctuations in financial markets. This approach suggests using the signals of indicators in technical analysis instead of using time series data which mostly cannot provide accurate results. For this purpose, using different indicators, 112 different signal models have been

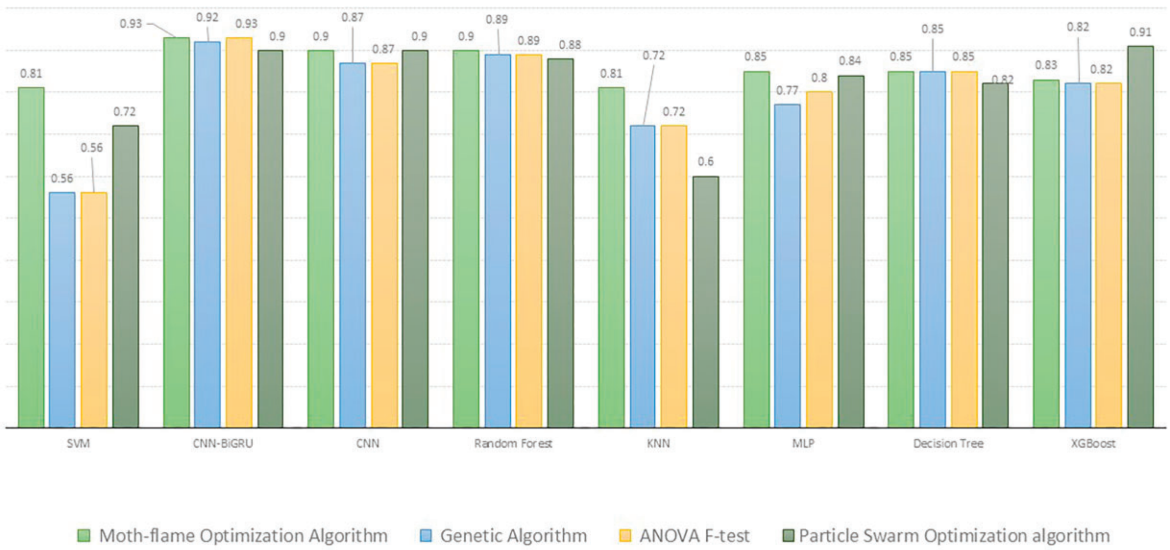

Figure 8: Comparison chart of different feature selections in terms of accuracy, precision, and recall.

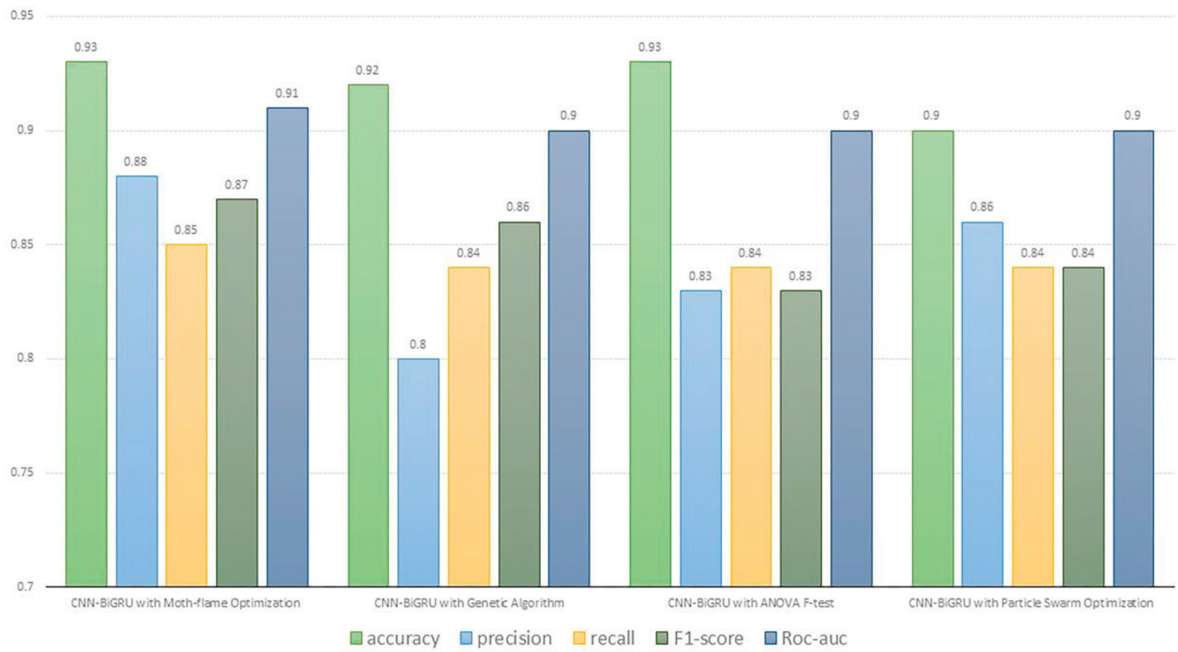

Figure 9: Trade-off between feature selection algorithms in the proposed model.

extracted. In the next step, using a hybrid DL model, price fluctuations have been predicted. Also, in order to optimize the prediction results, instead of using random search and grid search approaches, which are very time-consuming approaches, metaheuristic algorithms have been used. Finally, some other machine learning/DL algorithms and metaheuristic algorithms have been implemented and compared on the dataset. The results show that the proposed model is more accurate than other models.

The data used in this study have a wide range and variety improvement of the performance of the proposed approach. On the other hand, the identification of the most influential factors on the target variable by metaheuristic algorithms, the results of hybrid DL algorithms, and the hyperoptimization of its parameters by metaheuristic algorithms have led to very high accuracy in signal prediction. Creating our model with an accuracy of around 96% for predicting signals from technical analysis indicators makes investors lose money in the financial markets with very little error.

When investors use prediction modes like the Ichimoku cloud in the financial markets, the accuracy is very low, i.e., less than 10% on average. However, the proposed approach can correctly predict whether the prices will rise or fall with a probability of 49% (i.e., 36% accuracy). It can correctly predict the price decline when the price has also decreased with 13% accuracy. This concept shows that if investors invest an amount of about $100 in the global market of precious metals and use the Ichimoku cloud to identify the market trend, whether it is ascending or descending, to buy or sell with a probability of 51% in this market. Meanwhile, the proposed approach in this study has a 4% error for making wrong predictions. We showed that modeling using time series data simulate past data while providing a good accuracy on

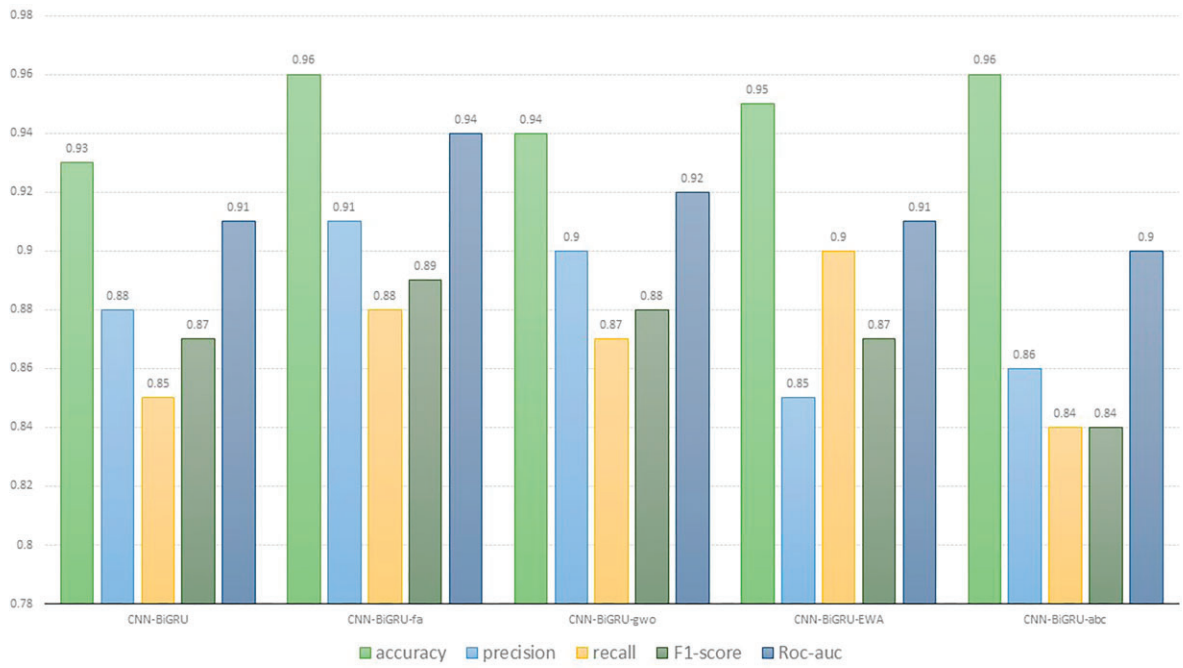

Figure 10: Results of optimized CNN-BiGRU algorithm.

future data. In addition to the fact that this approach can be used as a decision support tool for the gold global market, we encourage investors to apply this method in predicting other financial markets.

6. Conclusions and Future Works

Forecasting the financial markets and precious metals markets is very important for economic policymakers leading to an increase in social and economic stability. However, forecasting these markets is very difficult due to their volatile and complex nature. Therefore, the literature has seen several methods to improve the accuracy of forecasting models. One of them is the set of technical analysis indicators to increase investors' profitability in a short period.

Technical analysis indicators are popular tools among financial market traders. One of the main problems of these indicators is their high inaccuracy in producing buy or sell signals. In other words, the signals generated by these indicators have many errors. We can combine these indicators in different ways, and each of these analyzes can represent a signal to enter or leave the markets.

In this study, using MACD, Ichimoku, and MA, 112 signal models in the precious metals market were recorded. Each of these signal modes is an analysis model of indicators. In the Ichimoku indicator, when a cross occurs, the uptrend of the cloud can indicate a buy signal and the downtrend of the cloud is a sell signal where we can formulate it using a binary variable with the numbers 0 and 1. The literature shows that this type of analysis can issue a buy signal in 1040 records. Hence, the investor may suffer huge losses using this type of analysis.

This study proposed a model that can detect with a 94% probability to identify the signals correctly. First, to select the most influential variables on the target variable, we proposed a successful metaheuristic algorithm, i.e., MFO. After selecting the features, the presented approach is a hybrid of CNN and BiGRU. The hyperparameters of this algorithm are optimized using FA. We can say that a hybrid model of optimized NN for classifying fictitious signals is the main purpose of this paper. The proposed algorithm is compared with other DL and machine learning algorithms to measure its accuracy and validity. To evaluate the performance of the proposed algorithm in the feature selection, we used ANOVA *F*-test, GA, and PSO. The results show that the proposed algorithms can detect the signals very accurately.

Although this research helped academics and investors to create new indicators, there are many ways to extend our technical indicators in future research. For example, we can use data at the hourly, minute, and second intervals. With the expansion of features, we can apply different selection algorithms using multicriteria decision-making theories. Another important suggestion is that we can use different time series algorithms. Because financial markets have so much uncertainty and can be very difficult to predict, future researchers are encouraged to explore the role of the fuzzy approach and fuzzy data. Last but not least, we can consider other recent and powerful metaheuristic algorithms like lion-inspired optimization, red deer algorithm, and social engineering optimizer for the optimization of our model's parameters.

Conflict of interest statement

The authors confirm that there is no conflicts of interest.

References

Agudelo Aguirre A. A., Duque Méndez N. D., & Rojas Medina R. A. (2021). Artificial intelligence applied to investment in variable income through the MACD (moving average convergence/divergence) indicator. *Journal of Economics, Finance and Administrative Science*, **26**(52), 268–281. https://doi.org/10.1108/JEFAS-06-2020-0203.

Batten J. A., Lucey B. M., McGroarty F., Peat M., & Urquhart A. (2018). Does intraday technical trading have predictive power in precious metal markets? *Journal of International Financial Markets, Institutions and Money*, **52**, 102–113. https://doi.org/10.1016/j.intfin.2017.06.005.

Bhadoria A., & Marwaha S. (2020). Moth flame optimizer-based solution approach for unit commitment and generation scheduling problem of electric power system. *Journal of Computational Design and Engineering*, **7**(5), 668–683. https://doi.org/10.1093/jcde/qwaa050.

Chandar S. K. (2021). Hybrid models for intraday stock price forecasting based on artificial neural networks and metaheuristic algorithms. *Pattern Recognition Letters*, **147**, 124–133. https://doi.org/10.1016/j.patrec.2021.03.030.

Chandar S. K. (2022). Convolutional neural network for stock trading using technical indicators. *Automated Software Engineering*, **29**(1), 16. https://doi.org/10.1007/s10515-021-00303-z.

Chang C.-L., Ilomäki J., Laurila H., & McAleer M. (2020). Market timing with moving averages for fossil fuel and renewable energy stocks. *Energy Reports*, **6**, 1798–1810. https://doi.org/10.1016/j.egyr.2020.06.029.

Cheng T., Liu J., Yao W., & Zhao A. B. (2022). The impact of COVID-19 pandemic on the volatility connectedness network of global stock market. *Pacific-Basin Finance Journal*, **71**, 101678. https://doi.org/10.1016/j.pacfin.2021.101678.

Chong T. T. L., Ng W.-K., & Liew V. K.-S. (2014). Revisiting the Performance of MACD and RSI Oscillators. *Journal of Risk and Financial Management*, **7**(1), 1–12. https://doi.org/10.3390/jrfm7010001.

Chou J.-S., Chen L.-Y., & Liu C.-Y. (2023). Forensic-based investigation-optimized extreme gradient boosting system for predicting compressive strength of ready-mixed concrete. *Journal of Computational Design and Engineering*, **10**(1), 425–445. https://doi.org/10.1093/jcde/qwac133.

Chou J.-S., Nguyen N.-M., & Chang C.-P. (2022). Intelligent candlestick forecast system for financial time-series analysis using metaheuristics-optimized multi-output machine learning. *Applied Soft Computing*, **130**, 109642. https://doi.org/10.1016/j.asoc.2022.109642.

Deng S., Yu H., Wei C., Yang T., & Tatsuro S. (2021). The profitability of Ichimoku Kinkohyo based trading rules in stock markets and FX markets. *International Journal of Finance & Economics*, **26**(4), 5321–5336. https://doi.org/10.1002/ijfe.2067.

Díaz F., Henríquez P. A., & Winkelried D. (2022). Stock market volatility and the COVID-19 reproductive number. *Research in International Business and Finance*, **59**, 101517. https://doi.org/10.1016/j.ribaf.2021.101517.

Dingli A., & Fournier K. (2017). Financial time series forecasting—A deep learning approach. *International Journal of Machine Learning and Computing*, **7**, 118–122. https://doi.org/10.18178/ijmlc.2017.7.5.632.

El-Shorbagy M. A., & El-Refaey A. M. (2022). A hybrid genetic–firefly algorithm for engineering design problems. *Journal of Computational Design and Engineering*, **9**(2), 706–730. https://doi.org/10.1093/jcde/qwac013.

Elhoseny M., Metawa N., & El-hasnony I. M. (2022). A new metaheuristic optimization model for financial crisis prediction: Towards sustainable development. *Sustainable Computing: Informatics and Systems*, **35**, 100778. https://doi.org/10.1016/j.suscom.2022.100778.

Fischer T., & Krauss C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, **270**(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054.

Ghasemiyeh R., Moghdani R., & Sana S. S. (2017). A Hybrid Artificial Neural Network with Metaheuristic Algorithms for Predicting Stock Price. *Cybernetics and Systems*, **48**(4): 365–392. https://doi.org/10.1080/01969722.2017.1285162.

Gholizadeh H., Fathollahi-Fard A. M., Fazlollahtabar H., & Charles V. (2022). Fuzzy data-driven scenario-based robust data envelopment analysis for prediction and optimisation of an electrical discharge machine's parameters. *Expert Systems with Applications*, **193**, 116419.

Gillaizeau M., Jayasekera R., Maaitah A., Mishra T., Parhi M., & Volokitina E. (2019). Giver and the receiver: Understanding spillover effects and predictive power in cross-market Bitcoin prices. *International Review of Financial Analysis*, **63**, 86–104. https://doi.org/10.1016/j.irfa.2019.03.005.

Gunnarsson B. R., vanden Broucke S., Baesens B., Óskarsdóttir M., & Lemahieu W. (2021). Deep learning for credit scoring: Do or don't? *European Journal of Operational Research*, **295**(1), 292–305. https://doi.org/10.1016/j.ejor.2021.03.006.

Gurrib I., Kamalov F., & Elshareif E. (2020). Can the leading US energy stock prices be predicted using the Ichimoku cloud? *International Journal of Energy Economics and Policy*, **11**(1), 41–51. https://www.econjournals.com/index.php/ijeep/article/view/10260.

Houssein E. H., Abohashima Z., Elhoseny M., & Mohamed W. M. (2022). Hybrid quantum-classical convolutional neural network model for COVID-19 prediction using chest X-ray images. *Journal of Computational Design and Engineering*, **9**(2), 343–363. https://doi.org/10.1093/jcde/qwac003.

Islam M. R., Ali S. M., Fathollahi-Fard A. M., & Kabir G. (2021). A novel particle swarm optimization-based grey model for the prediction of warehouse performance. *Journal of Computational Design and Engineering*, **8**(2), 705–727.

Khatua A. (2016). An application of moving average convergence and divergence (MACD) indicator on selected stocks listed on National Stock Exchange (NSE). *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2872665.

Koo B., & Shin B. (2018). Applying novelty detection to identify model element to IFC class misclassifications on architectural and infrastructure Building Information Models. *Journal of Computational Design and Engineering*, **5**(4), 391–400. https://doi.org/10.1016/j.jcde.2018.03.002.

Kouaissah N., Orlandini D., Ortobelli S., & Tichý T. (2020). Theoretical and practical motivations for the use of the moving average rule in the stock market. *IMA Journal of Management Mathematics*, **31**(1), 117–138. https://doi.org/10.1093/imaman/dpz006.

Li J., Rao X., Li X., & Guan S. (2022). Gold and Bitcoin optimal portfolio research and analysis based on machine-learning methods. *Sustainability*, **14**, 14659.

Li X., Ma X., Xiao F., Xiao C., Wang F., & Zhang S. (2022). Time-series production forecasting method based on the integration of bidirectional gated recurrent unit (Bi-GRU) network and sparrow search algorithm (SSA). *Journal of Petroleum Science and Engineering*, **208**, 109309. https://doi.org/10.1016/j.petrol.2021.109309.

Liu H., & Long Z. (2020). An improved deep learning model for predicting stock market price time series. *Digital Signal Processing*, **102**, 102741. https://doi.org/10.1016/j.dsp.2020.102741.

Liu K., Zhao Z., Shi P., Li F., & Song H. (2022). Real-time surgical tool detection in computer-aided surgery based on enhanced feature-fusion convolutional neural network. *Journal of Computational Design and Engineering*, **9**(3), 1123–1134. https://doi.org/10.1093/jcde/qwac049.

Ma Y., Yang B., & Su Y. (2021). Stock return predictability: Evidence from moving averages of trading volume. *Pacific-Basin Finance Journal*, **65**, 101494. https://doi.org/10.1016/j.pacfin.2021.101494.

Maki D., & Ota Y. (2021). Impacts of asymmetry on forecasting realized volatility in Japanese stock markets. *Economic Modelling*, **101**, 105533. https://doi.org/10.1016/j.econmod.2021.105533

Mamoudan M. M., Mohammadnazari Z., Ostadi A., & Esfahbodi A. (2022). Food products pricing theory with application of machine learning and game theory approach. *International Journal of Production Research*, 1–21. https://doi.org/10.1080/00207543.2022.2128921.

Martia D., & Yasmine N. (2021). Indikator simple moving average dan relative strenght index untuk menentukan sinyal beli dan jual saham pada sektor infrastruktur. *Jurnal Pasar Modal Dan Bisnis*, **3**, 27–38. https://doi.org/10.37194/jpmb.v3i1.67.

Meher B., Hawaldar I., Cristi S., & Birau R. (2021). Forecasting stock market prices using mixed ARIMA model: A case study of Indian pharmaceutical companies. *Investment Management and Financial Innovations*, **18**, 42–54. https://doi.org/10.21511/imfi.18(1).2021.04.

Meng F., Song T., Xu D., Xie P., & Li Y. (2021). Forecasting tropical cyclones wave height using bidirectional gated recurrent unit. *Ocean Engineering*, **234**, 108795. https://doi.org/10.1016/j.oceaneng.2021.108795.

Mirjalili S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, **89**, 228–249. https://doi.org/10.1016/j.knosys.2015.07.006.

Moosavi J., Fathollahi-Fard A. M., & Dulebenets M. A. (2022). Supply chain disruption during the COVID-19 pandemic: Recognizing potential disruption management strategies. *International Journal of Disaster Risk Reduction*, **75**, 102983. https://doi.org/10.1016/j.ijdrr.2022.102983.

Moskowitz T. J., Ooi Y. H., & Pedersen L. H. (2012). Time series momentum. *Journal of Financial Economics*, **104**(2), 228–250. https://doi.org/10.1016/j.jfineco.2011.11.003.

Murphy J. J. (1999). *Technical analysis of the financial markets: A comprehensive guide to trading methods and applications*. Prentice Hall Press.

Niu D., Yu M., Sun L., Gao T., & Wang K. (2022). Short-term multi-energy load forecasting for integrated energy systems based on CNN-BiGRU optimized by attention mechanism. *Applied Energy*, **313**, 118801. https://doi.org/10.1016/j.apenergy.2022.118801.

Nor S. M., & Wickremasinghe G. (2017). The profitability of MACD and RSI trading rules in the Australian stock market. *Investment Management & Financial Innovations*, **11**, 194–199.

Pourkhodabakhsh N., Mamoudan M. M., & Bozorgi-Amiri A. (2022). Effective machine learning, meta-heuristic algorithms and multi-criteria decision making to minimizing human resource turnover. *Applied Intelligence*. https://doi.org/10.1007/s10489-022-04294-6.

Ray U., Banik D., Sarkar S., & Ganguly K. (2023). Deployment of random forest algorithm to prophesize the exact price of gold. *Interantional Journal of Scientific Research in Engineering and Management*. https://doi.org/10.55041/IJSREM17501.

Sulistiawan D., & Rudiawarni F. A. (2020). The use of moving average trading rules in Indonesian stock market. *International Journal of Business and Globalisation*, **24**, 139–150. https://doi.org/10.1504/IJBG.2020.104965.

Tripathi D. K., Chadha S., & Tripathi A. (2023). Metaheuristic enabled intelligent model for stock market prediction via integrating volatility spillover: India and its Asian and European counterparts. *Data & Knowledge Engineering*, **144**, 102127. https://doi.org/10.1016/j.datak.2022.102127.

Vaidya R. (2020). Moving average convergence-divergence (MACD) trading rule: An application in Nepalese stock market (NEPSE). *Quantitative Economics and Management Studies*, **1**, 366–374. https://doi.org/10.35877/454RI.qems197.

Xiao J., Wen F., Zhao Y., & Wang X. (2021). The role of US implied volatility index in forecasting Chinese stock market volatility: Evidence from HAR models. *International Review of Economics & Finance*, **74**, 311–333. https://doi.org/10.1016/j.iref.2021.03.010.

Xu D. (2022). Canadian stock market volatility under COVID-19. *International Review of Economics & Finance*, **77**, 159–169. https://doi.org/10.1016/j.iref.2021.09.015.

Yan M., Deng Z., He B., Zou C., Wu J., & Zhaoju Z. (2022). Emotion classification with multichannel physiological signals using hybrid feature and adaptive decision fusion. *Biomedical Signal Processing and Control*, **71**, 103235. https://doi.org/10.1016/j.bspc.2021.103235.

Zhang Y., Chu G., & Shen D. (2021). The role of investor attention in predicting stock prices: The long short-term memory networks perspective. *Finance Research Letters*, **38**, 101484. https://doi.org/10.1016/j.frl.2020.101484.

Zhang Y., Zhang Z., Yin T., & Liang W. (2022). Mathematical formulation and an improved moth-flame optimization algorithm for parallel two-sided disassembly line balancing based on fixed common stations. *Journal of Computational Design and Engineering*, **10**, 233–249. https://doi.org/10.1093/jcde/qwac134.

Zhong X., & Enke D. (2017). A comprehensive cluster and classification mining procedure for daily stock market return forecasting. *Neurocomputing*, **267**, 152–168. https://doi.org/10.1016/j.neucom.2017.06.010.

Zhou H., Bai J., Wang Y., Jiao L., Zheng S., Shen W., Xu J., & Yang X. (2021). Few-shot electromagnetic signal classification: A data union augmentation method. *Chinese Journal of Aeronautics*, **35**, 49–57.

Received: January 16, 2023. **Revised:** April 11, 2023. **Accepted:** April 12, 2023