

Received October 21, 2021, accepted November 8, 2021, date of publication November 11, 2021, date of current version November 19, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3127570

Can Deep Learning Improve Technical Analysis of Forex Data to Predict Future Price Movements?

MARCO FISICHELLA¹ AND FILIPPO GAROLLA²

¹L3S Research Center, Leibniz University of Hannover, 30167 Hannover, Germany

²SLR Engineering, 8010 Graz, Austria

Corresponding author: Marco Fisichella (mfisichella@L3S.de)

This work was supported in part by the European Commission for the eXplainable Artificial Intelligence in healthcare Management (xAIM) Project under Grant INEA/CEF/ICT/A2020/2276680.

ABSTRACT The foreign exchange market (Forex) is the world's largest market for trading foreign money, with a trading volume of over 5.1 trillion dollars per day. It is known to be very complicated and volatile. Technical analysis is the observation of past market movements with the aim of predicting future prices and dealing with the effects of market movements. A trading system is based on technical indicators derived from technical analysis. In our work, a complete trading system with a combination of trading rules on Forex time series data is developed and made available to the scientific community. The system is implemented in two phases: In the first phase, each trading rule, both the AI-based rule and the trading rules from the technical indicators, is tested for selection; in the second phase, profitable rules are selected among the qualified rules and combined. Training data is used in the training phase of the trading system. The proposed trading system was extensively trained and tested on historical data from 2010 to 2021. To determine the effectiveness of the proposed method, we also conducted experiments with datasets and methodologies used in recent work by Hernandez-Aguila *et al.*, 2021 and by Munkhdalai *et al.*, 2019. Our method outperforms all other methodologies for almost all Forex markets, with an average percentage gain of 20.2%. A particular focus was on training our AI-based rule with two different architectures: the first is a widely used convolutional network for image classification, i.e. *ResNet50*; the second is an attention-based network *Vision Transformer* (ViT). The results provide a clear answer to the main question that guided our research and which is the title of this paper.

INDEX TERMS Forex, expert advisor, genetic algorithm, metatrader, technical analysis, technical indicators, trading rules, trading system.

I. INTRODUCTION

Nowadays, people are more and more involved in trading currencies. Although cryptocurrencies promise better returns than foreign exchange, the foreign exchange (Forex) marketplace offers solid, extra secure and relatively regulated trading compared to cryptocurrency trading. For this reason, the Forex market has attracted quite a bit of interest from researchers in recent years. Various types of studies have been conducted to accomplish the task of accurately predicting future Forex currency rates.

Researchers have mainly focused on neural network models, pattern-based approaches and optimization techniques. The advent of artificial neural networks has played a major role in predicting forex currency rates. In particular, the

The associate editor coordinating the review of this manuscript and approving it for publication was Seyedali Mirjalili.

Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM) have been extensively studied for the prediction of temporal sequences.

The foreign exchange market is the world's largest market for trading foreign money, with a trading volume of over 5.1 trillion dollars per day. It is considered to be very complicated and volatile. Foreign exchange trading takes place around the clock (with the exception of the weekend), with buying and selling time divided into 4 basic time zones. Each of these zones has its own opening and closing hours. Forex trading is divided into three specific categories: Majors, Minors and Exotics. The most traded pairs in the Forex market are called majors.

For each Forex pair, we can retrieve the opening, high, low and closing prices. For security reasons, it is not possible for a single person to register and buy in the Forex market, but each individual must use third parties, such as brokers, who

are people or companies that have access to the Forex market and are able to buy or sell currencies. In the Forex market, there are only two alternatives: either buying currencies or selling them.

Predicting the Forex market has been a major goal of researchers in recent decades. There are two methods of predicting the market: fundamental and technical analysis. **Fundamental analysis** takes into account many factors, such as the financial system and political state of a country, the popularity of a company, all internal and external buying and selling news, etc. **Technical analysis** focuses on predicting future prices based solely on historical data. In this paper, we will focus on technical analysis.

Existing techniques are not suitable for all currency pairs and can give better results only for some randomly selected pairs in certain time frames. Moreover, most of the existing studies are evaluated only considering their prediction error. However, it is equally important to determine their impact on a real trading system where the decision to enter and invest in the market is as important as the decision to exit and disinvest.

In this article, we present our reusable trading system extensively trained on historical data measured in 4-hour intervals with candlesticks of 6 currency pairs: 4 major pairs such as GBP/USD, EUR/USD, USD/CHF and USD/JPY, and 2 minor pairs such as EUR/GBP and GBP/JPY. The entire dataset spans from 01/01/2010 to 04/30/2021, and we have conducted thorough experiments on the above currency pairs in three test series consisting of the entire year 2019, the entire year 2020, and the current time interval for the year 2021 (from 01/01 to 04/30) for all six currency pairs. For almost all currency pairs in the three long time intervals considered, our system has achieved stable and highly profitable results.

The remainder of the work is as follows. Section II presents numerous techniques that have been studied in recent years and can be classified into several categories: Regression strategies, optimization strategies, neural networks, etc. Section III introduces the basics of Forex, the trading system such as Meta Trader 5, as well as technical indicators and trading rules. Section IV explains our proposed trading system. Section V presents the results of experiments with historical data with candlesticks of 6 currency pairs on a long time frame extending from 01/01/2010 to 04/30/2021. We conclude our article by discussing the results and possible future directions in the last section.

II. RELATED WORK

In recent years, numerous hybrid techniques have been studied. Based on the works we reviewed, they can be classified into four categories: Regression strategies, Optimization strategies, Neural networks and others. These categories were formed according to the popularity of the main methods of forecasting system in recent years. In general, these techniques can give good results only for some randomly selected currency pairs in certain time frames. Pattern-based systems may contradict each other and propose opposite findings, such as the algorithm presented in [3] in contrast

to the approach proposed by [4]. Moreover, even if neural network-based approaches are promising, in most cases they are evaluated only by considering their prediction error. However, it is equally important to determine their impact on a real trading system where the decision to invest in the market is as important as the decision not to invest. In a few cases, such as the study conducted by Chandrinos *et al.* the investment return for an entire trading system was considered; however, their decision tree approach again beat the neural network in terms of total return.

A. REGRESSION METHODS

Raimundo and Okamoto [5] proposed a hybrid model for Forex prediction using wavelet models together with Support Vector Regression (SVR). They used a discrete wavelet transform (DWT) to interpret facts from their Forex dataset. In series, they used support vector regression (SVR) for prediction. They analysed the overall performance of their system with ARIMA and ARFIMA models. The effects confirmed that their system performed better than ARIMA and ARFIMA models.

Taveeapiradeecharoen *et al.* [6] proposed a version for time series testing and prediction based on compressed autoregression vectors. Initially, they used a random compression method to compress a large variety of Forex data into a smaller form. Then, they used Bayesian model averaging (BMA) approach to determine the loading of each randomly compressed datum and obtain the overlapping parameters. Their approach can provide out-of-sample forecasts up to fourteen days ahead of the current date. They concluded that their system is not suitable for forecasting all 30 foreign exchange currencies. Their proposed study outperformed the existing Bayesian autoregression benchmark for certain 6 currency pairs.

The authors of the study [7] proposed a variety of forecasting models by applying linear kernel SVR to historical data for the EUR/USD, GBP/USD and USD/JPY currency pairs from high frequency trading. They found a simple rule that produced high-quality results.

Support vector regression became the most commonly used approach in the papers we reviewed. Compressed vector autoregression, CRT regression tree and partial least squares regression were additionally used by the researchers. However, there are other algorithms such as lasso regression, logistic regression and multivariate regression which were abandoned in later years. The literature reviewed shows that the system based primarily on a regression model performs better than ARIMA and ARFIMA models [5], and model performance may additionally increase [8] when a regression model is combined with other techniques.

B. OPTIMIZATION TECHNIQUES

Chandrinos *et al.* [9] proposed a technical system for Forex trading using the Donchian channel method. The main reason was to create profitable portfolios for Forex buying and selling strategies. They first constructed the modified Renko bars (MRBs) by combining their trading rules. Their modified

MRBs proved to be more correct than the normal candlesticks used on Forex. They created an optimization layer used by eight currency pairs. Three global optimization algorithms were used: Hyperrectangle (Direct), Multilevel Coordinate Search (MCS) and Pity Beetle (PBA). They evaluated the performance primarily based on geometric return, arithmetic mean and Sharpe ratio. They found that the proposed version is not always suitable for three currency pairs, while for the others they achieved an overall return of 29% to over 200%.

Das *et al.* [10] proposed a hybrid approach combining deep learning machines for online sequential records and Krill Herd (KH). Krill Herd (KH) was dedicated to feature reduction. They used four foreign currency pairs. Their approach achieved the best RMSE. However, at MAE overall performance, their proposed model did not provide satisfactory results.

For optimising the approach to buy and sell foreign exchange, the authors of the article [11] used a genetic set of rules to develop a set of profitable buying and selling rules based entirely on the weighted moving average approach. They used four Forex pairs from 2000 to 2015. They used a time series of 4147 observations within a period of sixteen years from 2000 to 2015 and used the closing prices of four foreign currency pairs. The developed approach provided acceptably high returns on out-of-sample data. The rules obtained using the genetic algorithm resulted in significantly better returns than those obtained by exhaustive search.

In summary, these techniques are not suitable for all currency pairs and provide better results only for a few randomly selected pairs, as we can see in the proposed work.

C. NEURAL NETWORK

Ni *et al.* [12] proposed a model that predicts Forex time series using the C-RNN approach. C-RNN uses a convolutional neural network and a recurrent neural network. They used a statistically driven method to study the changing characteristics of Forex. They used records from the last 10 years to 2018 for nine currency pairs. Their dataset contained 2,000 of data points. Using convolutional neural network and long-term memory, they discovered that their proposed C-RNN version has much less error (calculated by RMSE) than LSTM and CNN.

Chandrinis *et al.* [9] proposed the Artificial Intelligence Risk Management System (AIRMS) based on machine learning. They developed two risk management structures: one using a neural network (AIRMS-ANN) and the other using a decision tree approach (AIRMS-DT). They used five pairs and categorized the output into two classes: profitable and non-profitable. When they categorized the output signal as profitable only, they obtained a profit increase of 50% over the version labeled with two categories. The F1 measure was used as the evaluation measure for both models. Both AIRMS-ANN and AIRMS-DT performed well on average and in some cases outperformed each other. When the portfolios were evaluated using the Kelly criterion, the decision tree again beat the neural network in terms of total return.

Dash [13] proposed a model using a higher order neural network forex prediction. They used a shuffled frog jump approach with the Pi-Sigma neural network to predict dynamic and nonlinear forex rates. Three currencies were used to apply their model. Their model provided higher accuracy combined with higher statistical power.

A recent study by Ahmed *et al.* [14] suggested that Forex rate prediction can be significantly improved by incorporating domain information into the system for training machine learning models. They used a long short-term memory model called FLF-LSTM. Using 10,078 4-hour candles of the EUR/USD pair, they showed that the proposed FLF-LSTM model had a 10.96% lower mean absolute error rate compared to the classical LSTM version.

Finally, the new study by Hernandez-Aguila *et al.* [1] presents a method for building forecasting models for the Forex market using multi-agent and fuzzy systems aimed at simulating the interactions that cause price changes. A comparison is also made with the architecture proposed by Munkhdalai *et al.* [2] which consists of seven neural networks with different activation functions, as well as a softmax layer and a multiplication layer with a skip connection, which are used to generate the dynamic weighting that decides which activation function is preferred. The results show that the method of Hernandez-Aguila *et al.* achieves prediction errors (using the mean absolute error) that are of the same order of magnitude as the errors obtained by Munkhdalai *et al.* and, in general, of models generated with Deep Learning, as well as of models generated with Random Forest, AdaBoost, XGBoost and Support Vector machines.

D. REMAINING METHODS

This is by no means an exhaustive list of all approaches, but we wanted to analyse the most recent work in the field and give the reader an overview of the other methods found in the literature.

The genetic algorithm and the SVM hybrid version were quite widely used strategies. The best feature of the SVM is that it can be used as a classifier [15] and as a regressor for prediction [16]. Nevertheless, a wrongly chosen kernel for the SVM can mean a big difference in the final result [16]. Moreover, some approaches depend on the choice of learning model, inputs and selection mechanisms.

Another perspective is pointed out by researchers who use chaos theory. Studies have demonstrated that the broad applicability of chaos theory can be used in broader ways. Lee [17] indicated that chaos theory can be successfully used for both economic time series prediction and trading strategy optimization. The problem was to determine the input parameters. The methods were chosen entirely according to the dynamics underlying the data and what type of analysis was intended for the system. This made the system tremendously complex and not always accurate.

Pattern-based techniques have received quite a bit of recognition. Pattern-based methods have also demonstrated their flexible adaptability. Some systems were able to switch the

statistics of predictors using the object properties that could be performed to design a selection of time series structures [18]. However, some pattern-based approaches were inconsistent: some systems worked well and offered excellent results with a given algorithm [3], while others did exactly the opposite [4]. Moreover, some pattern-based models were only able to predict changes over a short period of time and did not guarantee success for longer prediction periods [19].

Finally, the rest of the strategies included several types of methods used to predict the Forex market. It was found that Bayesian autoregressive trees (BART), Random Forest (RF), Naive Bayes (NB), ARIMA and many others were implemented and explored. Some of these algorithms have been explored individually, while some have been done in a hybrid fashion. Natural language processing has not been explored much, but techniques such as NLP, which are based on sentiment analysis and rely on news headlines, could easily be misled by using fake news [20]. Therefore, proper security measures must be taken for these methods.

III. PRELIMINARIES

A. TRADING IN FOREX

An order is primarily a request to make a transaction in order to open a position. Once a transaction is opened, it constitutes a position. Placing an order to close a position results in a trade in the opposite direction, e.g. if we originally bought, we now sell to close. When we open a buy position we are *long*, when we sell we are *short*.

To start a Forex trade, you have to open a position. There are different ways to open a position: either you buy the base currency and sell the quote currency (long) or you sell the base currency and buy the quote currency on the Forex (short). For example, in the GBP/USD currency pair, the GBP is the base currency and the USD is the quote currency.

A trader can use leverage when buying and selling. Leverage is the ratio that allows a small amount of money to trade a large amount. For example, if one trades 1000 dollars in GBP/USD with a leverage 1 : 30, the amount of the buy and sell transaction will be 30, 000 dollars instead of 1000 dollars. The profit or loss of a buy or sell transaction is calculated by subtracting the final value from the initial price of the Forex pair. Suppose the trader goes long with 30, 000 dollars in GBP/USD at a buy rate of 1.4060 dollars and closes his position at a sell rate of 1.4090 dollars. The difference is 0.0030 dollars, which is 20 pips. A pip is the smallest price movement an exchange rate can make according to market conventions. Since the initial position is a long position and the rate has risen, the transaction is profitable and the profit is $30,000 * 0.0030 = 90$ dollars. Consequently, the trader wants the price of the currency pair to rise when he is long and fall when he is short in order to make a profit.

B. TRADING SYSTEM: META TRADER 5

Meta Trader 5¹ is one of the most widely used trading platforms in the Forex market. It comes with a programming

¹<https://www.metatrader5.com/en/trading-platform>

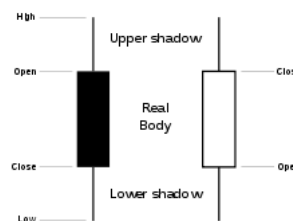


FIGURE 1. Candlesticks.

language, MQL5 [21], to implement automated trading programs. The final goal of our research is to provide the community with the implemented Expert Advisor² developed using MetaEditor as IDE (Integrated Development Environment) for MQL5 included in Metatrader 5, extending the common code written by Young.³

C. TECHNICAL INDICATORS

Technical analysis is the observation of past market movements with the aim of predicting future prices and dealing with the effects of market movement. Based on charts, analysis attempts to identify patterns in price charts. Basically, traders can visualize four types of charts: Bar chart, line chart, candlestick chart and figure chart. We have used candlestick charts.

In Fig. 1, some features of the candlestick are shown: i) the bar denotes the amplitude of the price between the opening and the closing, ii) two interconnected appendages called shadows representing the maximum and minimum price reached during the trading session.

A candle can have two colours: i) if the closing price is higher than the opening price, the candle is called *bullish* and has a *white* colour; ii) conversely, the candle is *black* and is called *bearish*.

Candlestick charts reveal patterns according to the amplitude of the body, the shape and the sequence of candles or groups of candles [22].

When the lower shadow is particularly long, the probability of reversal increases and the candle takes on a strong connotation of reversal. In fact, the longer the shadows, the greater the sense of instability in the market and the more likely the signal of reversal will be.

Technical indicators have been heavily used in Forex and there has been much research on how technical analysis and technical indicators can be used in predicting financial markets, with a focus on Forex. Many forecasting and trading methods have been proposed in the literature combining technical indicators with various machine learning models, with GA playing an important role. One of the earliest investigation was conducted in [23]. We also looked at the most recent and most cited papers in the last 20 years that use technical indicators and were studied

²https://github.com/marco82ger/AI_forex_fisichella_garolla

³<http://www.expertadvisorbook.com/MQL5BookSourceCode.zip>

in [22], [24]–[26]. In our trading system, the 20 most common and relevant technical indicators are used as the basis for trading rules. These technical indicators are: Adaptive Moving Average, Average Directional Moving Index, Donchian Channel, Stoller Bands, Bollinger Bands, Double Exponential Moving Average, Envelope Moving Average, Parabolic SAR, Fractal Adaptive Moving Average, Standard Deviation, Triple Exponential Moving Average, Average True Range, Bears Power, Bulls Power, MACD (Moving Average Convergence Divergence), Stochastic Oscillator, William' Percentage Range, Momentum, RSI (Relative Strength Index), and Heiken Ashi Candles.

D. TRADING RULES

We derive trading rules from each indicator listed in Section III-C. Specifically, a trading rule generates short and long signals. A signal is an indication to open a position in the market.

For example, Slow Stochastic may generate a short signal when the %K line crosses the %D line within the range iper-bought from high to low; similarly, Slow Stochastic may generate a long signal when the %K line crosses the %D line within the range iper-sold from low to high.

Trading rules can be more complex, such as rules calculated with dynamic channels as described in section III-D1.

1) TRADING RULES WITH DYNAMIC CHANNELS

Many operational strategies designed for short-term trading use signals provided by so-called *dynamic channels*. In some cases, the strategies are of the trend-following type, meaning that they aim to follow the main trend in the market; in other cases, they are instead of the reversal type, meaning that they try to take advantage of a possible trend reversal. The basic principle of these operation techniques is based on the individualization of some oscillation bands (envelopes):

- Containing for the majority of time the movement of prices;
- Concurring to characterize the trend followed by prices;
- Supplying interesting reversal signals.

These channels are dynamic in the sense that they measure the current instability in the market and adjust to the (cyclical) movements of prices, alternating between periods of consolidation (where instability decreases) and directional/impulsive periods (where instability increases). The technique of bands requires the identification of:

- A central reference average used to identify and exploit the primary trend of the market;
- A lower band and an upper band, which provide dynamic support and resistance respectively (as they move in line with the volatility observed in the market) and contain the price movement.

When volatility is low, the two bands narrow and move closer to prices (channel width decreases); when volatility

is high, the two bands widen and move away from prices (channel width increases).

We therefore begin by describing the most commonly used dynamic channels: Donchian Channel, Stoller Bands and Bollinger Bands.

a: DONCHIAN CHANNEL

One of the first operational techniques based on the use of channels was invented by Richard Donchian, who developed a trend-following system based on identifying the highs and lows of the last 20 days (the original system was based on Donchian's 4-week rule). Using this technique:

- Long positions are open when prices exceed the highest level in the last 20 days;
- Short positions are open when prices fall below the lowest level in the last 20 days.

With this technique, we look for the classic breakout signals that can initiate a directional move.

b: STOLLER BANDS

An experienced commodity trader, Manning Stoller, has developed an operational technique (Stoller Average Range Channels, STARC) that uses the concept of Average True Range (ATR). This differs from the simple daily range (the difference between the high and low prices) because it takes into account whether the trade is a gap-up or a gap-down. The ATR is the larger of these three differences:

- Today's maximum price minus today's minimum price;
- Today's maximum price minus yesterday's closing price;
- Yesterday's closing price minus today's low price.

To calculate Stoller Bands, a 6-period moving average is calculated to which the 15-period moving average of Average True Range multiplied by two is added (to get the upper band) or subtracted (to get the lower band). So there are three reference lines:

- The Upper Band equals to the 6-period moving average plus $(ATR * 2)$;
- The Central Average equals to the 6-period moving average;
- The Lower Band equals to the 6-period moving average minus $(ATR * 2)$.

While the technique proposed by Donchian is a trend-following type, the Stoller Bands is a reversal type. Specifically:

- Short signals occur when prices move above the upper band and draw a reversal short bar (e.g., a shooting);
- Long signals are obtained when prices fall below the lower band and draw a long reversal bar (for example, a hammer).

c: BOLLINGER BANDS

Bollinger Bands consists of three lines: a central moving average and two bands, one upper and one lower, which automatically widen and narrow depending on the volatility

expressed by the market. The calculation of Bollinger Bands according to the original formula of its creator is simple:

- For the upper band, we add to the central moving average (e.g. at 20 periods) the value of two times the standard deviation;
- For the lower band, we subtract from the central moving average the value of twice the standard deviation.

When the market is moving sideways, volatility decreases and the two bands tighten around prices. When the market is in a directional phase (either positive or negative), volatility increases and the two bands move away from prices.

The strategies that can be constructed using the Bollinger Bands are essentially two:

The Volatility Breakout: When the bands are compressed around prices, it means that a noticeable contraction of volatility is verified in the market, a situation that can anticipate the beginning of an impulsive movement, often confirmed bullish or bearish with a bar of the breakout. In this case, a trend-following strategy can be developed. For example:

- A long position can be opened when prices accelerate upwards after a breakout and move above the upper band. The signal is confirmed by both the breakout candle, which is above the upper band, and the next candle, which must not be bearish;
- A short position can be opened if prices fall quickly and below the lower band after the breakout. The signal is confirmed by both the breakout candle, which is below the lower band, and the next candle, which may not be bullish.

From a theoretical point of view, this is an interesting strategy, but it can lead to a late entry into the market when prices have already moved significantly.⁴

The Reversal Signals: The second technique exploits the exhaustion of a movement; it is based on the detection of a reversal configuration, which is verified when the courses come into contact with one of the two bands:

- A bearish signal occurs when price crosses above the upper band (the upper shadow of the first candle is above the upper band), but price then undergoes a correction and returns just below the band itself and the next session draws a bear candle;
- A bullish signal is signaled when price falls, drops below the lower band (the first candle has a lower shadow below the lower band), but then price bounces and returns above the same band, drawing a bull candle.

As an example, we report in Fig. 2 the signals short and long generated by this technique on USD/GBP chart.⁵

IV. THE TRADING SYSTEM

In this section, we present our trading system that uses the trading simulation layer and the genetic algorithm (GA) to



FIGURE 2. USD/GBP chart: signal generated by the signals of reversal technique on the Bollinger bands.

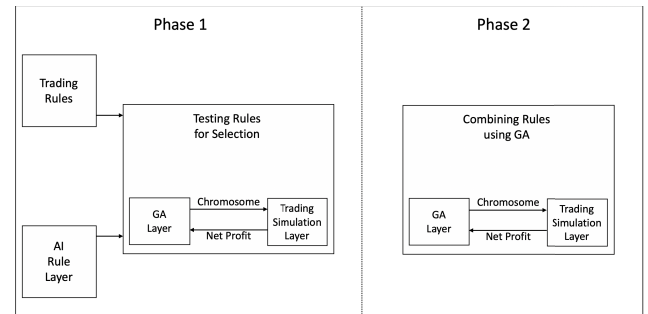


FIGURE 3. The framework of our trading system.

select and combine trading rules derived from both technical indicators, as presented in the previous section, and our AI layer, as described in section IV-A1, to generate buy/sell signals. The trading system uses time series price data of 6 currency pairs: 4 major pairs such as GBP/USD, EUR/USD, USD/CHF, USD/JPY and 2 minor pairs such as EUR/GBP and GBP/JPY.

The idea of combining trading rules comes from the real world of trading, where traders and analysts try to confirm their decision using different approaches and indicators before opening positions on the market.

Our proposed trading system consists of two phases: In the first phase, each trading rule, both AI-based rule and trading rules from technical indicators, is tested for selection; in the second phase, the profitable rules are selected from the qualified rules and combined.

An overview of the two phases of the proposed framework is shown in Fig. 3 and explained in the following subsections.

A. PHASE 1: EACH TRADING RULE IS TESTED FOR SELECTION

The goal of the first phase is to select trading rules, both AI-based rules and trading rules from technical indicators. Three levels are used: AI (section IV-A1), GA (section IV-A2) and the Trading Simulation (section IV-A3) layers.

The left part of Fig. 3 illustrates that both the trading rules from the technical indicators and the AI-based rule are given as input to the GA layer, which randomly generates chromosomes representing their genes with several parameter combinations that relate both directly to the specific trading rule (e.g., period *D* for the Slow Stochastic Oscillator -rule)

⁴In order to use signals of breakout it is preferable to use the Channel of Keltner, much more rapid in signalling the beginning of a directional trend.

⁵Source: Trading, G. Probo, Hoepfli.

as well as to the trade (e.g., the money management risk percentage, *risk%*, explained below). Then, all these rule candidates are simulated with different parameter combinations over the historical training data using the Trading Simulation layer, which gives as output the net profit, which is passed as a fitness value to the GA layer. Then, for each rule, the chromosome with the best performance in terms of net gain is selected. The rules whose best chromosome does not have a positive net gain are directly eliminated and do not enter the second phase.

1) AI-BASED RULE LAYER

Since experienced human traders can recognize patterns in the performance trends of currency pairs and make decisions based on the “look” of the performance charts, we wanted to investigate whether a computer vision-based AI agent could learn these patterns and successfully predict the optimal action. In an earlier work by [27], a comparison was made between three common neural network models, namely the multilayer perceptron, the CNN, and long short-term memory (LSTM). They found that a novel architecture based on CNNs outperformed the other models. For these reasons, we sought to take advantage of visualization and comprehensively improve the efficiency of intelligence analysis.

To isolate only the effect of the shape of the performance chart on action prediction, we generate images with no visible absolute value information. We also scale each y-axis of the graph to fill the image and best reveal the performance pattern. We provide 80 hours of historical context (each bar in the graph represents performance in a 4-hour stick) and ask the agent to predict the optimal action at the end of the visualized data window.

Specifically, each generated image is classified into three classes: (i) class *buy* if the best return prediction for a given time window (future 3 candles) is positive and larger than a threshold chosen in 700 points; (ii) class *sell* if the best return prediction for a given time window (future 3 candles) is negative and smaller than a threshold chosen in 700 points; (iii) class *stay* for all remaining images.

Since the number of *stay* samples vastly exceeded *buy* and *sell* samples, we subsampled 80% of the *stay* instances when creating the training dataset to obtain a balanced training dataset. The data distribution of the test dataset was not changed.

For the network model, we experimented with two different architectures. The first one is a widely used convolutional network for image classification, namely *ResNet50* [28]. The second is an attention-based network, namely *Vision Transformer (ViT)* [29].

Section V-D reports the details of the implementation and the results of the experiments.

2) THE GA LAYER: A GENETIC ALGORITHM FOR PARAMETER SELECTION

The unstable and chaotic structure of exchanges in the foreign exchange market complicates the analysis of forecasts. This

leads to the use of optimization methods. There are many heuristic methods such as genetic algorithms (GA), simulated annealing (SA) etc. to solve optimization problems. GA is one of the most popular heuristic optimization approaches that generates options that evolve over time. GA is based entirely on evolution and genetics. Heuristic strategies lead to near but not necessarily optimal solutions with reasonable computational effort and time.

A genetic algorithm is the heuristic algorithm that provides an acceptable answer to the problem in the majority of practically significant cases. Correctness of decisions is most often used for problems where the analytical solution is very difficult or even impossible.

GA contains concepts borrowed from nature. These are the concepts of heredity and variability. Heredity is the ability of organisms to pass on their traits and evolutionary characteristics to their offspring. Thanks to this ability, all living organisms pass on the characteristics of their species to their offspring.

The diversity of genes in living organisms ensures the genetic diversity of the population and is random because nature cannot know in advance which traits might be most useful in the future. This variability allows the appearance of living beings with new traits that can live under the new environmental conditions and pass on the new traits to their offspring.

In GA, there are two types of variations performed in the algorithm: (i) mutation, which is the variability that arises from the exploration of the search space; (ii) combination, which arises from the merging of genes using mating.

The *gene* is the atomic unit of information transmission. We can call a variable of function a gene. The gene is represented by a quantity: a real number. The set of gene variables of the trait under study is the characterizing feature of the *chromosome*.

In our approach, there are two types of chromosome representation: one associated with rules from technical indicators and one associated with AI-based rules. The chromosome representing technical indicator rules has two types of genes: the genes associated with the technical indicator and the genes associated with trading.

We give an example: the chromosome representation of the rule Slow Stochastic Oscillator is shown in table 1, where the period K , the period D and *Slowing* are the genes directly related to the oscillator, while the genes representing the interval of the Stop Loss, namely *intervalSL*, the multiplier factor of Take Profit with respect to Stop Loss, namely *takeProfitXSL*, the money management risk percentage, namely *risk%*, the trailing of the current price by a certain number of points, namely *trailingStop*, and the minimum profit for activating the trailing stop, namely *minProfit*, are the genes associated with the trade.

In contrast, the chromosome representing the AI-based rule has only genes associated with the trade, since all machine model parameters (e.g., model weights) are already computed during model training. As an example, the chromosome

TABLE 1. Chromosome representation of the rule slow stochastic oscillator.

K	10
D	3
Slowing	6
intervalSL	5
takeProfitXSL	2
risk%	2
trailingStop	500
minProfit	700

TABLE 2. GA parameter settings.

Number of chromosomes in the colony	100
Number of epochs without progress	50
Probability of mutation of each gene in %	5

contains only the *intervalSL*, *takeProfitXSL*, *risk%*, *trailingStop*, and *minProfit* genes.

All samples of the same evolution time are shuffled to form a population. Furthermore, the population is arbitrarily divided into two identical colonies: the parent colony and the offspring colony. In our work, the GA algorithm layer is implemented within the MetaTrader 5 platform.⁶ The following describes how the GA layer works:

- 1) First, value ranges are defined for each gene of a chromosome as a function of start-stop step values;
- 2) Second, the chromosomes representing the parameter combinations are randomly generated to form an initial proto-population;
- 3) Third, the fitness value is simulated and calculated by Trading Layer for each chromosome;
- 4) Finally, the main loop is executed by GA until the chosen number of iterations of the progeny is generated:
 - Preparing the population for reproduction after chromosome duplicates have been eliminated;
 - Isolation and protection of the reference chromosome (with the best fitness cost);
 - With each mating and mutation, new parents are added each time to prepare the population for the next era;
 - If the chromosome of the best offspring is higher than the reference chromosome, then we replace the reference chromosome.

We experimented with the values of various parameters to tune GA. For clarity, there are no universal parameter values, and it is good practice to assign them based on range. We varied the scale of the population, which ranges from 64 to 256, and the threshold for the number of epochs. We did not choose values that were too large, as this would not speed up the solution of the problem. As a result, we observed the following parameter settings for GA in our domain (Table 2):

The following optimization criteria were considered for GA and used with the fitness metric:

- Balance max - the highest value of balance.
- Net Profit max - the highest value of net profit.
- Expected Payoff max - the average return of a deal.
- Drawdown max - difference between the initial deposit and the minimum level below the initial deposit throughout the test period.
- Recovery Factor max — the highest value of the strategy's risk, i.e. the amount of money risked by Expert Advisor to achieve the obtained profit.
- Sharpe Ratio max — the highest value of efficiency and stability of a strategy.

For our GA layer, we decided to use net profit as the fitness metric. In summary, the GA layer outputs the best chromosomes with the largest fitness value determined.

3) TRADING SIMULATION LAYER

This layer is used to simulate arbitrary buy and sell rules on the given time series data with respect to specific currency pairs to generate buy/sell/Stop Loss/Take Profit signals (explained below) and calculate net profit and various statistics at the end of the simulation (number of trades/deals, total net profit, number of ticks, balance drawdown absolute/max/relative, consecutive profits, consecutive losses, etc.). Our trading simulation system takes a realistic approach to calculating net profit: we use a demo account with a trading broker⁷ simulating the placement of buy/sell positions. Our choice of a demo trading account is fundamental to the realistic calculation of the profit and loss of our proposed approach, while the net profit of a trading rule is used as a fitness value for the GA layer.

The simulation of placing a buy order is shown in algorithm 1. Similarly, a sell order is placed. For clarity, *Point* is the minimum amount of the price change.

First, a check is made to see if a new bar has been created. It is important to stick to the time period chosen in each case, e.g. 1-minute, 5-minute, 15-minute, hourly, daily and weekly bar chart. For our experiment, after experimenting with different time periods as described in section V-F, we chose 4-hour (4H). Second, we check whether a signal *BUY* has been triggered. If so, we check if there are any open sell positions and close them. Remember that closing a position results in a trade in the opposite direction, i.e. if we originally bought, we are now selling to close. So a *BUY* signal means we either close open *SELL* positions if there are any, or simply place a *BUY* order if we have no orders in the market. Then we check to see if there is another buy position that was previously opened that is priced within a close range of the current position. If this is the case, we do not open a new position, as this means opening a similar position twice.

At this point we calculate the trade size using the money management function, Stop Loss and Take Profit. Finally, we open the buy position. The following subsections explain the details of the latter calculations.

⁶<https://www.metatrader5.com/en/trading-platform>

⁷<https://www.icmarkets.com/en/open-trading-account/demo/>

Algorithm 1: Opening a BUY Position**Input:**

signal: trading rule BUY signal to open BUY position;
marketMovingHoriz: definition of Market moving horizontally;
symbol: the selected currency pair;
stopLoss: initial fixed Stop Loss in points;
takeProfitXSL: multiplier factor of Take Profit with respect to Stop Loss;
sellPosCount: the count of open SELL positions;
fixedVolume: volume in lots;
riskPercent: maximum risk per trade with respect to the current balance;

Output: A list of open positions

while OnTick() **do**

```

newBar = CheckNewBar();
if (newBar == true) then
  /* Open BUY position and close all
  SELL positions */
  if (signal == BUY) then
    /* Close open SELL positions */
    if (sellPosCount > 0) then
      CloseAllSellPosition();
    /* Do not open another BUY if
    there is a BUY position open
    with a price in a close range
    */
    if (currentBar.Close >
        (lastPriceBuyOpenPosition +
         (marketMovingHoriz/2)) || currentBar.Close <
         (lastPriceBuyOpenPosition -
         (marketMovingHoriz/2)) then
      /* Compute Stop Loss points */
      dynamicStopLoss = DynamicStopLoss();
      /* Stop Loss distance */
      SLDistance =
        MathMax(dynamicStopLoss, stopLoss);
      /* UseMoneyManagement */
      tradeSize = MoneyManagement(. . .);
      /* Take Profit is X times Stop
      Loss */
      buyProfit =
        BuyTakeProfit(takeProfitXSL, currentBar.Price);
    /* Buy */
    glBuyPlaced =
      TradeBuy(tradeSize, SLDistance, buyProfit);

```

Algorithm 2: Money Management Method**Input:**

MAXPERCENT: is a constant specifying the max risk and set to 10%;
symbol: the selected currency pair;
fixedVolume: default trade volume in lots;
riskPercent: maximum risk per trade with respect to the current balance;

SLDistance: Stop Loss distance in points;

Output: *tradeSize*: Calculated trade volume

if (*riskPercent* > 0) & (*stopLossDistance* > 0) **then**

if (*riskPercent* > *MAXPERCENT*) **then**

riskPercent = *MAXPERCENT*;

 margin = AccountInfoDouble(*accountBalance*) *
 (*riskPercent* / 100);

 tickSize = SymbolInfoDouble(*symbol*);

 tradeSize = (margin / *SLDistance*) / tickSize;

 tradeSize = VerifyVolume(*symbol*, tradeSize);

 return(tradeSize);

else

 tradeSize = *fixedVolume*;

 tradeSize = VerifyVolume(*symbol*, tradeSize);

 return(tradeSize);

per trade that results in a value between 2 and 4% of the current account balance. This value was calculated by our GA layer (more details in section IV-A2), where the risk value is represented as a gene (i.e. *risk%*) of a chromosome.

Algorithm 2 explains the method *MoneyManagement()* of Algorithm 1.

The *tradeSize* will contain our calculated trade volume. To begin, we check that *riskPercent* and *SLDistance* (Stop Loss distance) are both greater than zero. If both *riskPercent* and *SLDistance* are valid, then we proceed to calculate the trading volume.

Next, we calculate the amount of margin to risk by retrieving the account balance using the *AccountInfoDouble()* function. Then we determine the tick size and store it in *tickSize*. The tick size represents the profit or loss resulting from the movement of a single point.

b: STOP LOSS AND TAKE PROFIT

The trader specifies the number of points for the Stop Loss and Take Profit, which are then calculated for the order from the current price.

Stop Loss: For a buy order, the Stop Loss is calculated by subtracting the number of points set for the Stop Loss from the opening price, while for a sell order we add this amount. First, the symbol's point value is multiplied by the Stop Loss value. For example, a Forex symbol with five digits after the decimal point has a point value of 0.00001 dollars. If we set a Stop Loss of 500 points, we multiply 500 by 0.00001 to get a value of 0.005 dollars. This value is then added or subtracted to the opening price to determine the Stop Loss price.

a: MONEY MANAGEMENT TO CALCULATE TRADE SIZE

Money management is a process for optimally adjusting position size in relation to risk. One approach to calculating trade size is to look at the distance between the current price and the proposed Stop Loss. We have calculated the optimal risk

Take Profit: As a Stop Loss price is calculated, the Take Profit price is also calculated, just in reverse order.

A Stop Loss or Take Profit must have a minimum distance from the current opening price. The minimum distance is called the *stop level* and is retrieved from the broker server.

Instead of setting a fixed Stop Loss in points, there are also dynamic techniques to define it, such as using an indicator value or a technical support/resistance level. The *DynamiyStopLoss()* function in algorithm 1 retrieves the lowest low (since we are opening a buy position, it would otherwise be the highest high) of the last 5 bars and assigns the result to the *dynymicStopLoss* variable. For clarity, the cheque of the last 5 bars is a parameter computed by our GA layer, where the interval value is represented as a gene (i.e. *intervalSL*) of a chromosome. Similarly, we calculated the gene *takeProfitXSL* as the multiplier factor of Take Profit with respect to the stop loss, which gives a value of 2.

We conclude this section by introducing the trailing stop as a stop loss that moves by a certain number of points as the price moves. We can delay a trailing stop by requiring a minimum profit to be reached, which is specified in our algorithm as the variable *minProfit*. Both *trailingStop* and *minProfit* values are computed by our GA layer as values of genes of a chromosome; the best output was found with *trailingStop* equal to 450 points and *minProfit* equal to 700 points.

So far, we have presented a fixed trailing stop, but we have also traded dynamically according to an indicator, such as the Parabolic Stop and Reverse (PSAR) indicator. In our system, the use or nonuse of dynamic trailing with PSAR indicator was assessed by the GA layer of phase 2 using a binary gene in the chromosome.

c: CALCULATION OF THE TOTAL NET PROFIT

The final step of the trading simulation level is to calculate the total net profit. For each open position that is closed, either due to a reversal signal triggered by our system or due to the Stop Loss or Take Profit being reached, a profit or loss is calculated. The sum of all profits/losses in the entire time frame under consideration for a given currency pair represents our total net profit.

B. PHASE 2: SELECTING AND COMBINING PROFITABLE RULES

In the first phase, for each trading rule based on both technical indicators and AI, we obtain the best chromosome with the best fitness value. The rules whose best chromosomes do not have a positive net profit are directly eliminated and do not enter the second phase, where the qualified rules are combined into a trading system. Two layers are used to select the combined trading rules: the GA layer and again the Trading Simulation layer, which was already used in the first phase.

The right part of Fig. 3 illustrates how, at the beginning, the GA layer takes the trading rules selected in the first phase and then GA generates random chromosome candidates. It is important to highlight that the chromosomes in this phase are of a different typology than in the first phase, as each

TABLE 3. Chromosome representation of a candidate rule combination.

Rule 1	Rule 2	Rule 3	...	Rule N
0	1	0	...	1

chromosome represents a different rule selection solution; more details are shown in section IV-B1. The output of this layer consists of candidate chromosomes sent to the Trading Simulation layer, which simulates their behaviour using the historical training data. The latter layer returns as output the net profit as a fitness value to the layer GA. This process runs in a loop until the best chromosomes with the highest net profit are found.

1) THE GA LAYER: A GENETIC ALGORITHM FOR COMBINATION OF TRADING RULES

It is important to emphasise that the GA layer in this phase is different from that of the first phase. Here, the layer GA is used to combine the qualified rules from the first phase into a trading system.

The following describes how this layer works:

- 1) In primis, each chromosome consists of the same number of genes as the number of qualified rules from the first phase. Each gene is binary, that is, it is represented by either 0 or 1, and indicates whether the trade rule was selected to be present in the combination. An example of the representation of a chromosome with N trading rules is shown in Table 3. In this example, the total number of combinations is 2^N ;
- 2) The layer GA randomly generates chromosome candidates with the goal of forming an initial population;
- 3) For each of the generated chromosomes, the subset of trade rules to be present in the combination is selected. For such a subset of trading rules in combination, GA calculates the average values of several parameters: the interval of the Stop Loss, *intervalSL*; the multiplier factor of Take Profit with respect to Stop Loss, *takeProfitXSL*; the money management risk in percent, *risk%*; the trailing of the current price by a certain number of points, *trailingStop*; the minimum profit for activating the trailing stop, *minProfit*. These values are the genes associated with the output of the trading rule of each chromosome from the first phase;
- 4) Third, the fitness value for each chromosome is calculated and sent from the Trading Layer;
- 5) GA goes into the same loop as reported in section IV-A2.

The fitness metric is the net profit for the GA level; its parameters are the same as those presented in section IV-A2. This process loops until the best chromosomes with the highest net gain are found.

Finally, this stage outputs the best chromosome with the best fitness value found so far, which is the final rule combination.

2) TRADING SIMULATION LAYER: COMPUTING THE NET PROFIT OF THE RULES COMBINATIONS

This is the final layer that combines the qualified trading rules extracted from the chromosomes to generate buy/sell signals for open positions that resemble a unique trading ensemble rule and form the combined trading system. A hard voting method is used in this layer. The following describes the operation of this layer:

- 1) To begin, the layer stores the values of the *intervalSL*, *takeProfitXSL*, *risk%*, *trailingStop*, and *minProfit* genes;
- 2) Then, for each chromosome the Trading Simulation layer computes the average of these four genes for the enabled rules in the chromosome, i.e., the rules whose gene is set to 1;
- 3) To open a position, we consequently use an ensemble approach with statistical mode: we open a buy/sell position if this is the most frequent signal sent by the trading rules active in the chromosome, just like a hard voting classifier. Then the net profit is given as an output;
- 4) The rest of the steps performed by this layer are the same as reported in section IV-A3

V. EXPERIMENTS

All our experiments are reproducible and the code of our fully implemented Expert Advisor is publicly available.⁸

A. INVESTMENT SETUP

In our experiments, we used the following configuration:

- Our investment was limited to 10,000 dollars;
- We allocated for each transaction the amount resulting from the money management risk outlined in section IV, with a cap of 4% of the current available capital;
- Leverage was set at 1 : 30;
- Was chosen ideal execution for order placement with zero latency during trade execution;
- A demo account with a trading broker⁹ was used to simulate the placement of long/short positions.

B. EVALUATION METRICS

The following evaluation metrics were calculated for our experiments:

- **Net Profit:** the financial result (profit/loss) of all trades in money.
- **Total Trades:** the total number of trades.
- **Profit Factor:** the ratio of gross profit to gross loss. Gross profit is the sum of all profitable trades, while gross loss is the sum of all losing trades.
- **Expected Payoff:** the average return of one deal.

TABLE 4. Test-set details.

Currency Pairs	2019	2020	2021
	Ticks	Ticks	Ticks
EUR/USD	14,107,144	22,493,176	6,418,430
USD/CHF	13,057,137	15,598,601	4,256,816
USD/JPY	16,510,745	19,068,209	4,487,949
EUR/GBP	15,810,439	19,224,884	5,481,037
GBP/JPY	32,155,222	35,299,958	9,181,770
GBP/USD	20,904,274	24,012,594	7,117,293

- **Drawdown in %:** the difference between the initial deposit and the minimum level below the initial deposit throughout the test period.
- **Recovery Factor:** the value reflects the riskiness of the strategy, i.e. the amount of money Expert Advisor will risk to achieve the profit made.
- **Sharpe Ratio:** the ratio of the arithmetic average profit for the position holding period to the standard deviation thereof.

C. DATASETS

The data was collected and represents the historical data of 6 currency pairs measured in 4-hour intervals: 4 major pairs such as GBP/USD, EUR/USD, USD/CHF, USD/JPY and 2 minor pairs such as EUR/GBP and GBP/JPY. The data set covers the period from 01-01-2010 to 30-04-2021 and the training set was created using data from 01-01-2010 to 31-12-2018. The remaining data forms the test set. The data was extracted from Metatrader 5 software presented in section III-B. The experiments with all 6 currency pairs were conducted on three test sets:

- Full year 2019: from 01-01-2019 to 31-12-2019;
- Full year 2020: from 01-01-2020 to 31-12-2020;
- Year 2021: from 01-01-2021 to 30-04-2021.

Trading rules are active on every candle, but once a position is open and in the market, our approach checks every tick of the market. A tick is an event marked by a new price for a symbol; ticks are more common than candlesticks. In table 4 we give the number of ticks for each interval of the test series.

D. AI-BASED RULE LAYER

In this section, we explain our experimental results for the AI-based rule layer.

1) EXPERIMENTAL SETUP

As anticipated in section IV-A1, we experimented with two different architectures for the network model: a convolutional-based image classification network, i.e. ResNet50, and an attention-based network, i.e. Vision Transformer (ViT). The models were trained on a GPU instance Google Colab (NVIDIA K80 GPU) with the Adam optimizer [30] with a weight penalty of 0.0001 using Tensorflow¹⁰ and Keras.¹¹

⁸https://github.com/marco82ger/AI_forex_fisichella_garolla

⁹<https://www.icmarkets.com/en/open-trading-account/demo/>

¹⁰<https://www.tensorflow.org/about/bib>

¹¹<https://github.com/keras-team/keras>

TABLE 5. Network hyperparameters used in ViT.

Image size	224, 224, 3 (RGB)
Patch size	16, 16
Projection dimension	64
Number of transformer layers	8
Number of attention heads	4
Multilayer perceptron neurons in transformer layers	[128, 64]
Neurons in fully connected layers	[2048, 1024]

The ResNet50 network was initialized with weights pre-trained on the ImageNet [31] dataset and fine-tuned on our dataset. Images fed into the network are first resized to the input shape of 224×224 pixels, without preserving the aspect ratio. They are then converted to RGB format and each color channel is centered with respect to the ImageNet dataset without scaling.

The ViT network was initialized with random weights and trained directly on our dataset. The network has an input size of 224×224 pixels and each image is then split into non-overlapping patches of 16×16 pixels. The images fed to the network are first resized to the input size without maintaining the aspect ratio, and then normalized to a mean of zero and a standard deviation of one. The normalization parameters are calculated based on the training dataset. We used Adam's recommended default values for the hyperparameters β_1 and β_2 . The model weights are randomly initialized as suggested in [32]. Table 5 lists the network hyperparameters used in ViT.

We generate a validation dataset by randomly selecting 10% of the training samples. These samples are removed from the training dataset and used to evaluate the model during training. We used a batch size of 64 and trained for 100 epochs, evaluating the model at the end of each epoch using the validation dataset. At the end of training, the model weights with the least validation loss are recovered and evaluated against a retained test dataset.

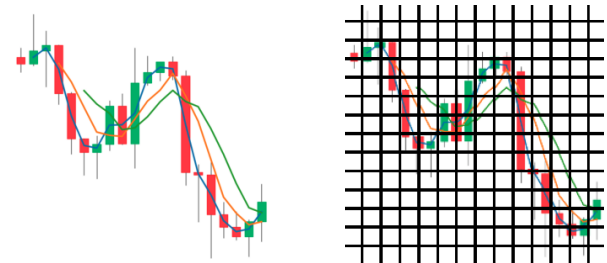
The trained model is deployed for inference in a Docker container running Tensorflow Serving docker image.¹² Integration with Metatrader is achieved by introducing a proxy server between Metatrader and the Tensorflow Serving server. The proxy server, implemented in Python, receives historical information over the last 20 time units and uses MPLFinance¹³ to generate an image visualization of the data similar to how the model's training dataset was generated (candle stick chart with running averages). This image is then passed to the Tensorflow model to obtain a classification. The resulting prediction is finally returned to the Metatrader Expert Advisor.

2) DATASET

The same data presented in section V-C were used for training. The training set was created using data from 01/01/2010 to 12/31/2018. The remaining data were reserved for testing and validation sets.

¹²<https://www.tensorflow.org/tfx/serving/docker>

¹³<https://github.com/matplotlib/mplfinance>



(a) Sample generated image. (b) Sample patches used by ViT.

FIGURE 4. Sample images as input of the AI models.

Using the MPLFinance library, a series of candlestick plots were created in *Yahoo style* showing a configurable number of data points before the classification point. The selected time window is 20 data points, covering a total of 80 hours of history per plot. The images are generated at 100 dpi and each generated image is 519×516 pixels. In addition to the candlesticks themselves, the images also show the moving average curve in time steps of 2, 4 and 6 past bars. An example of a generated image is shown in Fig. 4a.

For illustration, Fig. 4b shows an example of the patches used by the ViT architecture: We show the patches extracted from the original RGB image and separated by a black border. The network operates on the images after normalizing them to a mean of zero and a standard deviation of one, and has no border between the patches.

Each image is classified into three classes: (i) class *buy* if the best return prediction for a given time window is positive and larger than a threshold chosen in points; (ii) class *sell* if the best return prediction for a given time window is negative and smaller than a threshold chosen in points; (iii) class *stay* for all remaining images. In our experiments we investigated several time windows and thresholds, as reported in the following section V-D3.

The historical data is very unbalanced, as class *stay* significantly outperforms the other two classes. To address this issue, we subsampled the stay class for the training and validation datasets by randomly discarding 80% of the samples. This created a balanced dataset that is used to train our models. To measure performance in practice, we did not subsample the test dataset.

3) RESULTS

We explored different settings for class labels by varying the time window for future observation, i.e., the number of candles to be observed (namely, k), and the threshold in points according to the definition that each image is classified into three classes: (i) class *buy* if the best return prediction for a given time window (k) is positive and greater than a threshold (Points); (ii) class *sell* if the best return prediction for a given time window (k) is negative and less than a threshold (Points); (iii) class *stay* for all remaining images.

For each of the pairs of settings (Points/ k), we have given the best accuracy on the test set in Figure 5. As you can see, the best performance is obtained when the time window is

Points/k	1	2	3	4	Average
300	0.39	0.41	0.42	0.41	0.41
500	0.41	0.42	0.44	0.41	0.42
700	0.49	0.53	0.61	0.59	0.56
900	0.49	0.55	0.59	0.57	0.55
Average	0.45	0.48	0.51	0.50	

FIGURE 5. Accuracy for several pairs of settings (Points/k). Points denotes the threshold in points and k indicates the number of candles to be observed i the future time window.

TABLE 6. Accuracy of our networks.

	ViT	ResNet50
Train Accuracy	0.932	0.823
Validation Accuracy	0.416	0.397
Test Accuracy	0.504	0.607

TABLE 7. Confusion matrices of our networks.

Actual	ViT			ResNet50			
	Predicted			Predicted			
	Buy	Sell	Stay	Actual	Buy	Sell	Stay
Actual Buy	36	34	55	Actual Buy	0	62	63
Actual Sell	34	47	68	Actual Sell	2	64	83
Actual Stay	556	785	1474	Actual Stay	26	978	1811

equal to 3 future candles and the threshold is set to 700 points. It can be observed that looking at some candles in the future to classify images gives more stability to the model; moreover, increasing the boundary of the class stay avoids current fluctuations for the model decision. For the remaining results, we choose k equal to 3 and a threshold of 700 points.

Considering that the test dataset is very unbalanced (the class stay accounts for 91% of the data), the accuracy is not a good indicator of the actual performance of the model, but for clarity we report it in Table 6.

We use the confusion matrices to make inferences about the performance of the models in the test data set. The results are listed in table 7.

Training with the balanced dataset shows that none of the networks succeed in learning an effective prediction strategy. When analysing the image data, it is shown that the performance trend alone is not sufficient to predict the optimal behaviour. Table 8 shows some examples of ambiguous images that appear very similar but belong to opposite classes, preventing our networks from learning a generalised model.

E. RESULTS ON PHASE 1

The goal of the first phase is to select trading rules: Rules whose best chromosomes do not have a positive net profit are eliminated directly and do not advance to the second phase. Table 9 lists the qualified and non-qualified trade rules.

F. RESULTS WITH DIFFERENT TIME PERIODS

We also conducted our experiments with different time periods: 1- hour (1H), 4-hour (4H), 12-hour (12H), daily (1D). The main goal is to select the best profitable time interval.

TABLE 8. Examples of ambiguous images that appear very similar but belong to opposite classes.

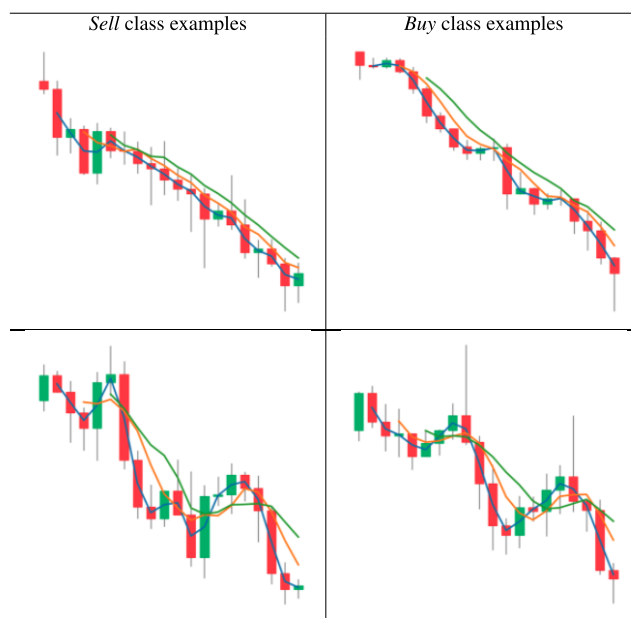


TABLE 9. Unqualified and qualified trading rules.

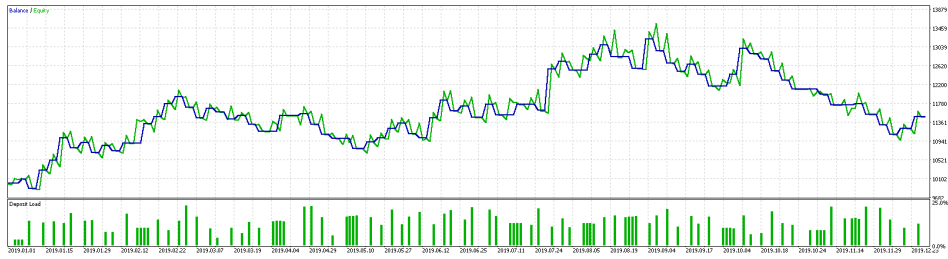
Rule Name	
Unqualified Rules	Qualified Rules
Adaptive Moving Average, Donchian channel, Stoller Bands, Double Exponential Moving Average, Envelope Moving Average, Fractal Adaptive Moving Average, Triple Exponential Moving Average, Avarage True Range, MACD (Moving Average Convergence Divergence), William' Percentage Range, Momentum, RSI (Relative Strength Index), AI-based.	Average Directional Moving Index, Bollinger Bands, Standard Deviation, Parabolic SAR, Bears Power, Bulls Power, Stochastic oscillator, Heiken Ashi Candles.

For each experiment, we reported the net profit and the total number of trades. As the results presented in table 10 show, the best period is the 4-hour period (4H), which outperforms the others, in some cases with peak values more than 10 times better.

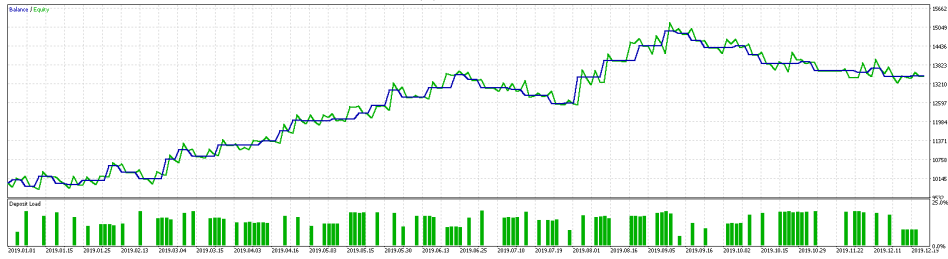
G. RESULTS ON PHASE 2

In the second phase, the profitable rules are selected from the qualified rules and combined. The final output is the best chromosome with the highest net profit. The best chromosome is obtained by combining the following rules: Bollinger Bands, Stochastic Oscillator, Bears Power, Bulls Power and Heiken Ashi Candles. Using this combination we implemented the Expert Advisor for trading.

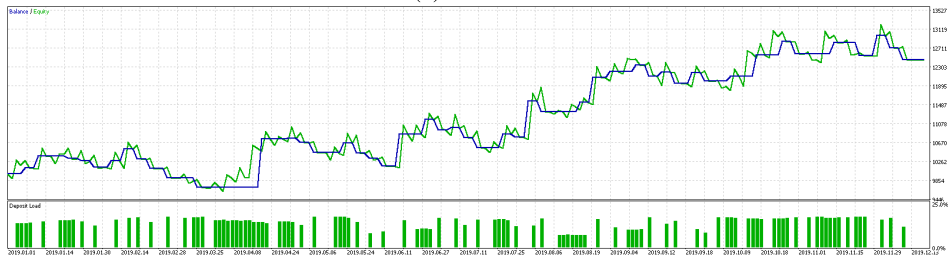
The results of experiments with all metrics presented in section V-B are shown in table 11. For almost all currency pairs, we are profitable during the three long time intervals considered and the results show that the best strategy is to



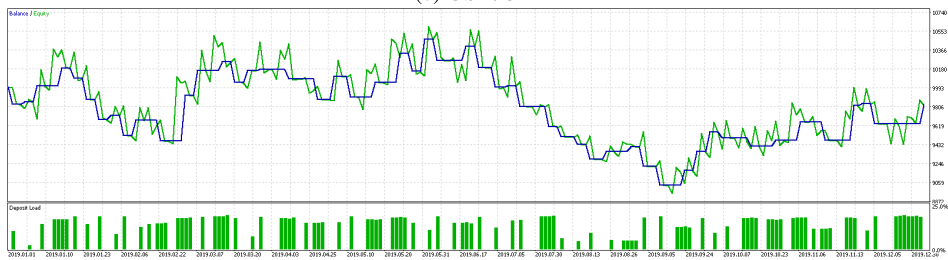
(a) GBP/USD



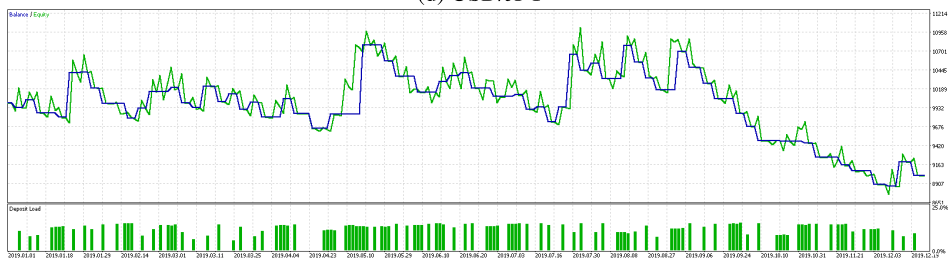
(b) EUR/USD



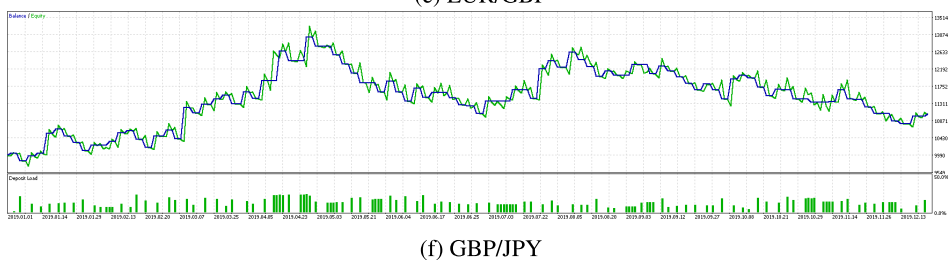
(c) USD/CHF



(d) USD/JPY

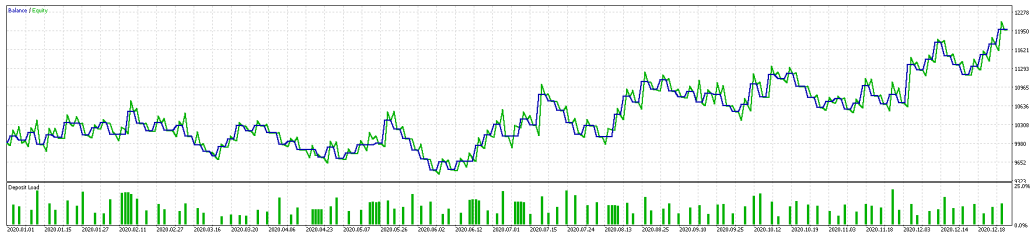


(e) EUR/GBP

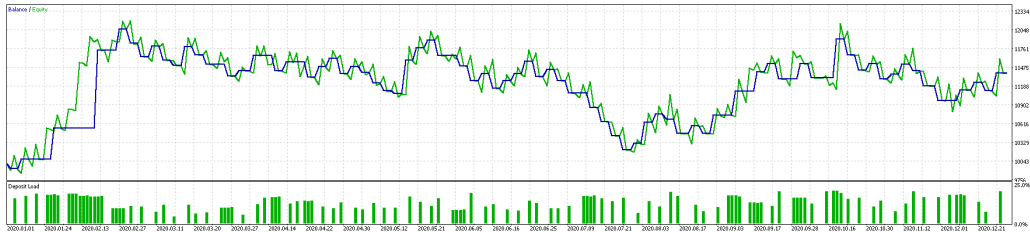


(f) GBP/JPY

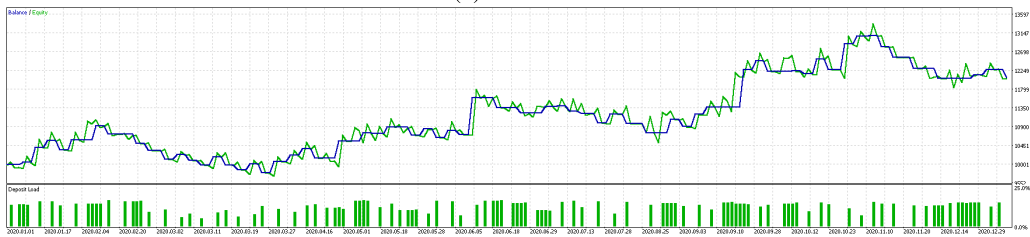
FIGURE 6. Balance/equity graphs and deposit load for the entire 2019.



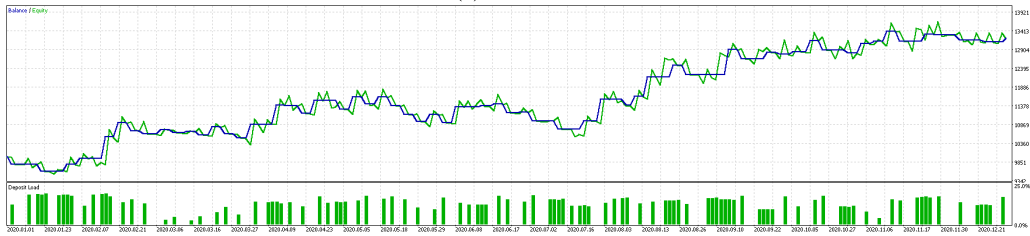
(a) GBP/USD



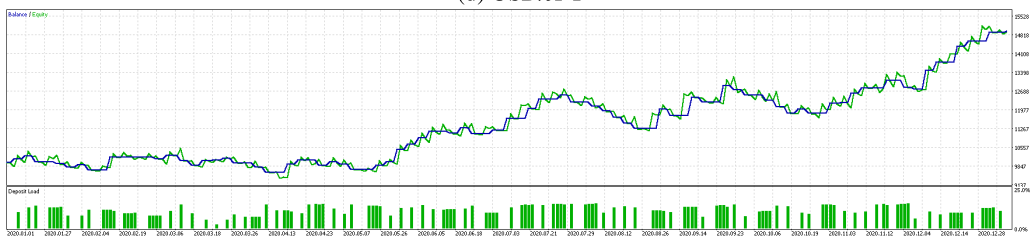
(b) EUR/USD



(c) USD/CHF



(d) USD/JPY

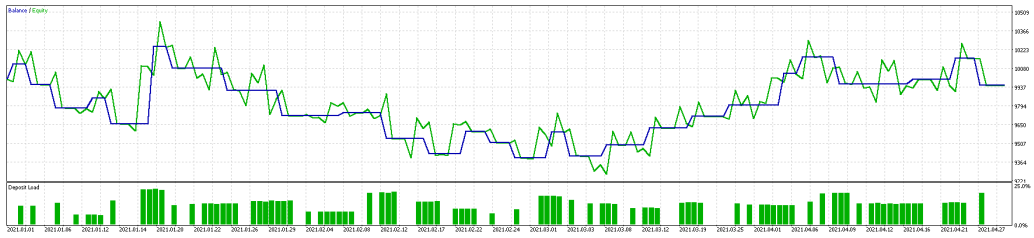


(e) EUR/GBP

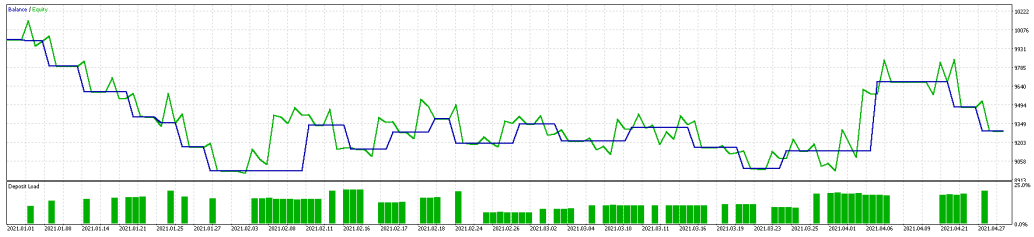


(f) GBP/JPY

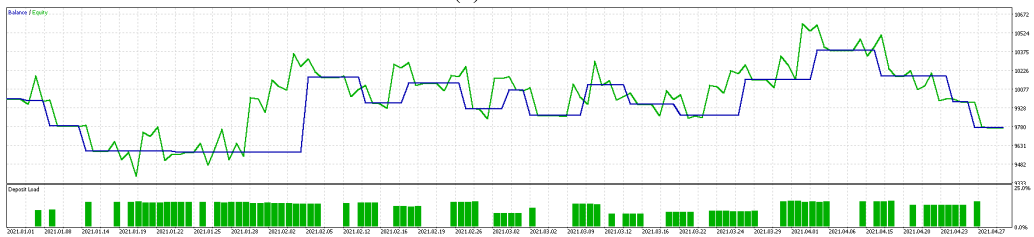
FIGURE 7. Balance/equity graphs and deposit load for the entire 2020.



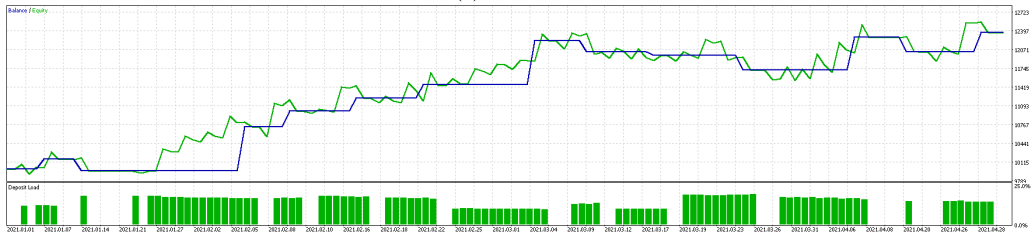
(a) GBP/USD



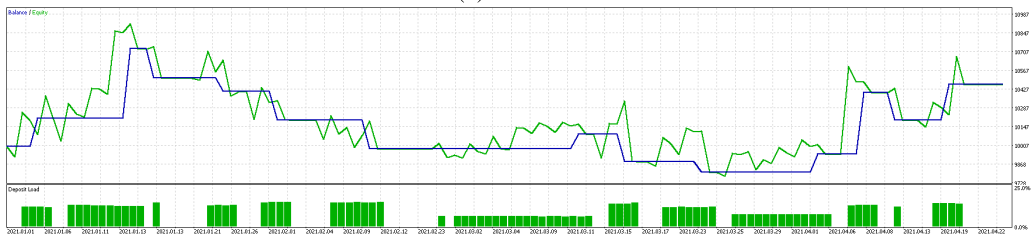
(b) EUR/USD



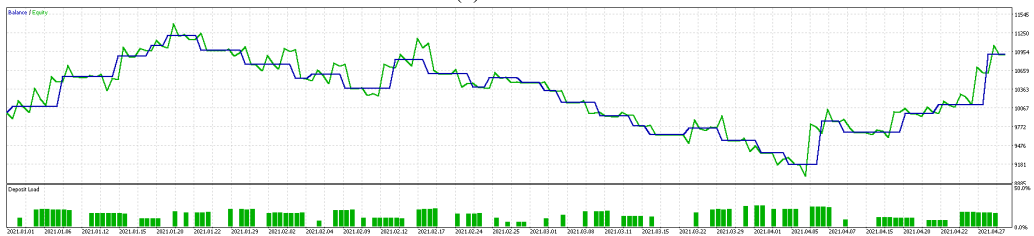
(c) USD/CHF



(d) USD/JPY



(e) EUR/GBP



(f) GBP/JPY

FIGURE 8. Balance/equity graphs and deposit load for 2021, from 01.01 until 30.04.

TABLE 10. Experiments with multiple data sets with different time periods: 1- hour (1H), 4-hour (4H), 12-hour (12H), daily (1D). In green the best net profit, in red the worst.

Date	Currency Pairs	1H		4H		12H		1D	
		Net Profit	Trades	Net Profit	Trades	Net Profit	Trades	Net Profit	Trades
2019 (01.01. - 31.12.)	GBP/USD	385.77	150	1490.90	74	-25.67	27	0	0
	EUR/USD	2113.09	59	3470.01	46	4185.46	15	222.55	1
	USD/CHF	-207.01	62	2466.51	47	813.95	15	0	0
	USD/JPY	-647.39	68	-175.52	51	293.97	18	0	0
	EUR/GBP	1420.55	66	-986.34	62	124.78	17	-71.12	1
	GBP/JPY	190.53	196	117.27	89	-13.41	33	-305.18	4
Yearly Total		3255.54		6382.83		5379.08		-153.75	
2020 (01.01. - 31.12.)	GBP/USD	-718.31	213	3883.91	102	860.66	32	-36.06	1
	EUR/USD	194.61	138	884.04	78	653.88	27	0	0
	USD/CHF	1060.37	85	1172.22	68	1125.94	25	7.02	3
	USD/JPY	620.94	89	1552.46	66	-91.34	26	123.32	1
	EUR/GBP	-463.25	103	4526.89	71	203.92	19	-80.22	3
	GBP/JPY	417.40	248	3438.72	97	49.69	32	-18.69	4
Yearly Total		1111.76		15458.24		2802.75		-4.63	
2021 (01.01. - 30.04.)	GBP/USD	1717.38	55	-48.82	27	-244.08	9	0	0
	EUR/USD	-390.54	27	-714.86	21	759.29	9	0	0
	USD/CHF	588.77	24	-260.77	20	168.01	5	61.04	1
	USD/JPY	109.46	22	2301.49	13	-203.12	8	0	0
	EUR/GBP	-218.89	20	1368.48	14	-140.90	4	0	0
	GBP/JPY	-2286.38	64	900.31	29	222.83	7	0	0
Yearly Total		-480.20		3545.83		562.03		61.04	

TABLE 11. Experiments on several datasets. In green the net profit, in red the net loss.

Date	Currency Pairs	Metrics						
		Net Profit	Total Trades	Profit Factor	Expected Payoff	Drawdown in %	Recovery Factor	Sharpe Ratio
2019 (01.01. - 31.12.)	GBP/USD	1490.90	74	1.19	20.15	19.18	0.57	0.09
	EUR/USD	3470.01	46	1.87	75.43	12.83	1.78	0.28
	USD/CHF	2466.51	47	1.54	52.48	9.91	2.33	0.18
	USD/JPY	-175.52	51	0.96	-3.44	15.56	-0.11	-0.01
	EUR/GBP	-986.34	62	0.85	-15.91	20.53	-0.44	-0.05
	GBP/JPY	117.27	89	1.01	1.32	19.34	0.05	0.02
Yearly Total		6382.83						
2020 (01.01. - 31.12.)	GBP/USD	3883.91	102	1.44	38.08	11.66	2.79	0.16
	EUR/USD	884.04	78	1.12	11.33	18.13	0.40	0.06
	USD/CHF	1172.22	68	1.18	17.24	12.82	0.73	0.08
	USD/JPY	1552.46	66	1.29	23.52	11.70	1.24	0.11
	EUR/GBP	4526.89	71	1.77	63.76	12.45	2.95	0.23
	GBP/JPY	3438.72	97	1.50	35.45	11.22	2.53	0.17
Yearly Total		15458.24						
2021 (01.01. - 30.04.)	GBP/USD	-48.82	27	0.98	-1.81	11.11	-0.04	0.00
	EUR/USD	-714.86	21	0.67	-34.04	11.60	-0.61	-0.15
	USD/CHF	-260.77	20	0.87	-13.04	9.61	-0.27	-0.05
	USD/JPY	2301.49	13	3.58	177.04	7.02	2.63	0.55
	EUR/GBP	1368.48	14	2.06	97.75	8.81	1.42	0.30
	GBP/JPY	900.31	29	1.30	31.05	20.99	0.38	0.12
Yearly Total		3545.83						

invest in all currency pairs simultaneously to reduce the risk of loss and remain profitable.

For clarity, we also plot the investment curve over time for all 6 currency pairs performed with the three test sets: Fig. 6 for the year 2019, Fig. 7 for the year 2020, and Fig. 8 for the year 2021. Specifically, we report the Balance/Equity chart and the deposit load chart.

The balance shows the amount of deposited money in our trading account. The profit/loss of our orders is added to or subtracted from our trading account balance when we close our orders. Balance and equity show the same values

as long as there are no open orders. When we have open orders on our trading account, the balance does not change and remains unchanged, but the equity is adjusted according to the profit/loss of our current orders.

The deposit load value shows the percentage of the account balance that is used to open positions. In our experiment the deposit load was always below 25%.

H. COMPARISON WITH OTHER STRATEGIES

To determine the effectiveness of the proposed method, we conducted experiments with datasets and methodologies

TABLE 12. Comparison of our strategy in terms of revenue (%) with other methods. In bold the best result.

Strategy	GBP/USD	EUR/USD	USD/CHF	USD/JPY	EUR/GBP	GBP/JPY	USD/CAD	AUD/USD
Buy and hold	-58.00%	-11.94%	-31.09%	-247.35%	+6.73%	-1.96%	+8.97%	-5.71%
Munkhdalai et al. like	+4.39%	0.68%	0.95%	-6.74%	-	-	-3.88%	+3.49%
Hernandez-Aguila et al.	+3.44%	+1.37%	+0.56%	+278.59%	-	-	+10.00%	+2.38%
Ours	+38.83%	+8.84%	+11.72%	+15.52%	+45.27%	+34.39%	-5.35%	+12.55%

used in recent work by Hernandez-Aguila *et al.* [1] and by Munkhdalai *et al.* [2]. In their work, daily exchange rates are used for the following Forex markets: GBP/USD, EUR/USD, USD/CHF, USD/JPY, USD/CAD, and AUD/USD. The 2020 prices are used as a validation dataset.

Table 12 shows a comparison of our method with three baselines: the two methods by Hernandez-Aguila *et al.* and by Munkhdalai *et al.* and a buy-and-hold strategy. In the buy-and-hold strategy, we subtract the closing price reached at the end of 2020 from the closing price at the beginning of the year, which corresponds to a buy order opened on the earlier date and closed on the later date. In the table, the best results are highlighted in bold. We have also reported our results for EUR/GBP and GBP/JPY to ensure consistency with our previous experiments and dataset. As you can see, our method outperforms all other methods for almost all Forex markets in terms of revenue.

Moreover, this confirms that most existing studies evaluated only considering their prediction error are not sufficient to ensure an effective trading methodology. In fact, the decision to enter and invest in the market is as important as the decision to exit and disinvest.

VI. CONCLUSION AND FUTURE WORK

In this paper, a complete trading system for Forex time series data is developed using a combination of trading rules. The idea of combining trading rules comes from the real world of trading, where traders and analysts try to confirm their decision using different approaches and indicators before investing in the market. We provide the community with the implemented trading system developed for MQL5 using MetaEditor as IDE (Integrated Development Environment).

The implementation of the system is done in two phases: In the first phase, each trading rule, both AI-based rule and trading rules from technical indicators, is tested for selection, and in the second phase, profitable rules are selected and combined among the qualified rules.

A special emphasis was put on training our AI-based rule with two different architectures: the first one is a widely used convolutional network for image classification, i.e. *ResNet50*; the second is an attention-based network *Vision Transformer* (ViT). The results provide a clear answer to the main question that guided our research: *Can Deep Learning improve technical analysis of Forex data to predict future price movements?* At present, the answer is negative. In general, neural network-based approaches are most often evaluated by considering only their prediction error, without taking into account the overall impact on a broader trading system where

the decision to open a position and invest is as important as the decision to close the position. We have gone beyond this limitation and considered a real functioning trading system.

To determine the effectiveness of the proposed method, we conducted experiments with datasets and methodologies used in recent work by Hernandez-Aguila *et al.* [1] and by Munkhdalai *et al.* [2]. Our method outperformed all other methods for almost all Forex markets in terms of revenue. In addition, the proposed trading system has been extensively trained on historical data measured in 4-hour intervals with candlesticks of 6 currency pairs: 4 major pairs such as GBP/USD, EUR/USD, USD/CHF, USD/JPY, and 2 minor pairs such as EUR/GBP and GBP/JPY. The entire dataset spans from 01/01/2010 to 04/30/2021, and we conducted thorough experiments on the above currency pairs in three test series consisting of the entire year 2019, the entire year 2020, and the current time interval for the year 2021 (from 01/01 to 04/30). For almost all currency pairs during the three long time intervals considered, our system has provided stable and highly profitable results with a low risk profile. Since it is not easy in finance to construct a stable mechanism that always generates profits, we conclude that the best strategy is to invest in all currency pairs simultaneously to reduce the risk of loss and remain profitable.

In future work, we would like to investigate how deep learning algorithms behave when we use them not only for technical analysis, but in combination with fundamental analysis, which takes into account several factors such as the financial system, the political situation of a country, all internal and external buying and selling news, etc. In addition, various methods can be explored to combine trading rules beyond the hard voting approach proposed in this paper.

REFERENCES

- [1] A. Hernandez-Aguila, M. García-Valdez, J.-J. Merelo-Guervós, M. Castañón-Puga, and O. C. López, "Using fuzzy inference systems for the creation of Forex market predictive models," *IEEE Access*, vol. 9, pp. 69391–69404, 2021.
- [2] L. Munkhdalai, T. Munkhdalai, K. Park, H. Lee, M. Li, and K. Ryu, "Mixture of activation functions with extended min-max normalization for Forex market prediction," *IEEE Access*, vol. 7, pp. 183680–183691, 2019.
- [3] P. Taveeapiradeecharoen and N. Aunsri, "Dynamic model averaging for daily Forex prediction: A comparative study," in *Proc. Int. Conf. Digit. Arts, Media Technol. (ICDAMT)*, Feb. 2018, pp. 321–325.
- [4] A. V. Contreras, A. Llanes, A. Pérez-Bernabeu, S. Navarro, H. Pérez-Sánchez, J. J. López-Espín, and J. M. Cecilia, "ENMX: An elastic network model to predict the FOREX market evolution," *Simul. Model. Pract. Theory*, vol. 86, pp. 1–10, Aug. 2018.
- [5] M. S. Raimundo and J. Okamoto, "SVR-wavelet adaptive model for forecasting financial time series," in *Proc. Int. Conf. Inf. Comput. Technol. (ICICT)*, Mar. 2018, pp. 111–114.

- [6] P. Taveeapiradeecharoen, K. Chamnongthai, and N. Aunsri, "Bayesian compressed vector autoregression for financial time-series analysis and forecasting," *IEEE Access*, vol. 7, pp. 16777–16786, 2019.
- [7] C. Serjam and A. Sakurai, "Analyzing predictive performance of linear models on high-frequency currency exchange rates," *Vietnam J. Comput. Sci.*, vol. 5, no. 2, pp. 123–132, May 2018.
- [8] S. Achchab, O. Bencharef, and A. Ouabar, "A combination of regression techniques and cuckoo search algorithm for FOREX speculation," in *Recent Advances in Information Systems and Technologies*, Á. Rocha, A. M. Correia, H. Adeli, L. P. Reis, and S. Costanzo, Eds. Cham, Switzerland: Springer, 2017, pp. 226–235.
- [9] S. K. Chandrinis, G. Sakkas, and N. D. Lagaros, "AIRMS: A risk management tool using machine learning," *Expert Syst. Appl.*, vol. 105, pp. 34–48, Sep. 2018.
- [10] S. R. Das, D. Mishra, and M. Rout, "An optimized feature reduction based currency forecasting model exploring the online sequential extreme learning machine and krill herd strategies," *Phys. A, Stat. Mech. Appl.*, vol. 513, pp. 339–370, Jan. 2019.
- [11] S. Galeshchuk and S. Mukherjee, "Forex trading strategy optimization," in *Decision Economics: In the Tradition of Herbert A. Simon's Heritage*, E. Bucciarelli, S.-H. Chen, and J. M. Corchado, Eds. Cham, Switzerland: Springer, 2018, pp. 69–76.
- [12] L. Ni, Y. Li, X. Wang, J. Zhang, J. Yu, and C. Qi, "Forecasting of Forex time series data based on deep learning," *Proc. Comput. Sci.*, vol. 147, pp. 647–652, Jan. 2019.
- [13] R. Dash, "Performance analysis of a higher order neural network with an improved shuffled frog leaping algorithm for currency exchange rate prediction," *Appl. Soft Comput.*, vol. 67, pp. 215–231, Jun. 2018.
- [14] S. Ahmed, S.-U. Hassan, N. R. Aljohani, and R. Nawaz, "FLF-LSTM: A novel prediction system using Forex loss function," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 106780.
- [15] B. J. de Almeida, R. F. Neves, and N. Horta, "Combining support vector machine with genetic algorithms to optimize investments in Forex markets with high leverage," *Appl. Soft Comput.*, vol. 64, pp. 596–613, Mar. 2018.
- [16] T. N. T. Thu and V. D. Xuan, "Using support vector machine in FoRex predicting," in *Proc. IEEE Int. Conf. Innov. Res. Develop. (ICIRD)*, May 2018, pp. 1–5.
- [17] R. S. T. Lee, "COSMOS trader—chaotic neuro-oscillatory multiagent financial prediction and trading system," *J. Finance Data Sci.*, vol. 5, no. 2, pp. 61–82, Jun. 2019.
- [18] E. Bartoš and R. Pinčák, "Identification of market trends with string and D2-brane maps," *Phys. A, Stat. Mech. Appl.*, vol. 479, pp. 57–70, Aug. 2017.
- [19] A. Wilinski, "Time series modeling and forecasting based on a Markov chain with changing transition matrices," *Expert Syst. Appl.*, vol. 133, pp. 163–172, Nov. 2019.
- [20] A. Ceroni, U. K. Gadiraju, and M. Fischella, "Improving event detection by automatically assessing validity of event occurrence in text," in *Proc. 24th ACM Int. Conf. Inf. Knowl. Manage.*, J. Bailey, A. Moffat, C. C. Aggarwal, M. de Rijke, R. Kumar, V. Murdock, T. K. Sellis, and J. X. Yu, Eds., Melbourne, VIC, Australia, Oct. 2015, pp. 1815–1818.
- [21] A. R. Young, *Expert Advisor Programming for MetaTrader 5*. Nashville, TN, USA: Edgehill, 2013.
- [22] M. Ozturk, I. H. Toroslu, and G. Fidan, "Heuristic based trading system on Forex data using technical indicator rules," *Appl. Soft Comput.*, vol. 43, pp. 170–186, Jun. 2016.
- [23] R. J. Sweeney, "Beating the foreign exchange market," *J. Finance*, vol. 41, no. 1, pp. 163–182, Mar. 1986.
- [24] P. Lipinski, *Evolutionary Strategies for Building Risk-Optimal Portfolios*. Berlin, Germany: Springer, 2008, pp. 53–65.
- [25] N. Jin, E. Tsang, and J. Li, "A constraint-guided method with evolutionary algorithms for economic problems," *Appl. Soft Comput.*, vol. 9, no. 3, pp. 924–935, Jun. 2009.
- [26] S. Deng, K. Yoshiyama, T. Mitsubuchi, and A. Sakurai, "Hybrid method of multiple kernel learning and genetic algorithm for forecasting short-term foreign exchange rates," *Comput. Econ.*, vol. 45, no. 1, pp. 49–89, 2015.
- [27] Y.-C. Tsai, J.-H. Chen, and J.-J. Wang, "Predict Forex trend via convolutional neural networks," *J. Intell. Syst.*, vol. 29, no. 1, pp. 941–958, Sep. 2018.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, pp. 1–9, Jun. 2015.
- [29] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," *CoRR*, vol. abs/2010.11929, pp. 1–22, Oct. 2020.
- [30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017, *arXiv:1412.6980*.
- [31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [32] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, Y. W. Teh and M. Titterton, Eds., Rome, Italy, vol. 9, May 2010, pp. 249–256.



MARCO FISICHELLA received the Ph.D. degree (*magna cum laude*) from Leibniz University, Hannover, Germany. He has more than 13 years of experience in artificial intelligence both in industry and academia. He was a Lecturer of the artificial intelligence course for the master's degree in computer science with Leibniz University. From 2015 to March 2021, he was the Head of data at Otto Group, where he led the department responsible for implementing algorithms against online fraud. Since April 2021, he has been leading the L3S Research Centre, Leibniz University, and the research group with a focus on artificial intelligence and intelligent systems. Within the scope of the activities at L3S, new methods of artificial intelligence are to be explored both in the field of basic research and in application-oriented research, especially for the application fields of mobility, intelligent production, and personalized medicine. He led various AI and data science projects for corporate clients, namely Otto, Vodafone, DriveNow, and Bonprix. His research interests include artificial intelligence, machine learning, data mining, information retrieval, generative models, event detection, and clustering methods based on statistical approaches and near duplicate detection.



FILIPPO GAROLLA received the master's degree in computer science and engineering from the Politecnico di Milano University of Technology. He is currently an Expert in computer vision and machine learning at SLR Engineering. He was specialized in the development of computer vision solutions for industrial and road management applications.

...