Int. J. Advanced Design and Manufacturing Technology, 2025, Vol. 18, No. 4, pp. 11-24

DOI: *ISSN: 2252-0406* *https://admt.isfahan.iau.ir*

A New Method for Generating Trading Signals in Financial Markets by a Combination of Deep Learning and Reinforcement Learning

Ahang Golabi, Mojtaba Salehi *, Hossein Nahid Titkanloo

Department of Industrial Engineering
Payame Noor University, Tehran, Iran
E-mail: ahang.golabi@pnu.ac.ir, Mojtaba.salehi@pnu.ac.ir, Hossein_Nahid@pnu.ac.ir
*Corresponding author

Received: 20 May 2025, Revised: 29 June 2025, Accepted: 5 August 2025

Abstract: The present study offers a new approach for identifying trading signals in the financial market using machine learning techniques. It employs graphical and correlational research methods, utilizing support learning techniques in MATLAB to test hypotheses. The study proposes an automated trading system combining reinforcement learning and deep learning to determine trade signals and position sizes. The framework combines an LSTM network with Q-learning, an out-of-policy reinforcement learning algorithm. Q-learning aims to maximize overall reward by learning from actions deviating from the existing policy. Our study introduces a new framework that utilizes the collective intelligence of multiple expert traders to learn across different time frames. It shows that using Fundamental and technical indicators independently or in combination to train LSTMs for predicting currency movements in Forex significantly improves prediction accuracy. The study introduces a third class to represent small changes in currency pair prices between two consecutive days, improving prediction accuracy. It also describes a new method for determining the most suitable threshold value to define the unchanged class. Additionally, the study trains LSTMs to predict values k days into the future, and searches for the influence of varying training iterations on accuracy values.

Keywords: Financial Market, Machine Learning, Trading Signals

Biographical notes: Ahang Golabi received her BSc in Industrial Engineering from Azad University of Sanandaj. She received her MSc in Information Technology Management from Payame Noor University and is currently a PhD candidate in industrial engineering at this university. **Mojtaba Salehi** received his PhD in Industrial Engineering from Tarbiat Modares University. He is currently an Assistant professor at Payame Noor University. His field of research is mainly Financial Engineering, System dynamics, Machine Learning & Data Mining and Applied Operation Research. **Hossein Nahid Titkanloo** received his PhD in Industrial Engineering from the Payame Noor University. Also, he is currently an Assistant Professor at Payame Noor University. His field of research is mainly industrial engineering, artificial intelligence and System Analysis.

Research paper

1 INTRODUCTION

Stock prices fluctuate daily due to various factors, including politics. This unpredictability complicates trading strategies for buying or selling stocks. Stock market analysis focuses on optimizing portfolios [1], identifying investment strategies [2], and assessing risks [3]. Forecasting stock price trends has attracted interest from various fields, including economics, finance, statistics, and machine learning [4-6].

Financial trading has been extensively studied, leading to the development of various methods. These approaches are classified into two classes: modern and traditional. Traditional ones are technical [2] and fundamental analysis [3]. Modern trading methods include algorithmic trading [4] and machine learning (ML) [3]. Abundant data, limitations of traditional methods, and the complicated nature of financial markets emphasize the need for strategies like algorithmic trading.

Studies have proposed algorithms for anticipating stock movements using machine learning (ML) techniques like support vector machines (SVM) and artificial neural networks (ANN) [5-8]. Researchers are using deep learning approaches like LSTM and RNN [9-10] for predicting stock prices. They also use trading algorithms to maximize profits [3]. Algorithmic trading can use rule-based strategies or ML approaches. Rule-based strategies are derived from traditional or mathematical models, while ML-based trading involves training computers on historical data to operate autonomously. This method offers several advantages over traditional trading.

Algorithmic trading, which utilises machine learning, offers advantages over rule-based trading. ML-based algorithms have the capability of extracting information and patterns from historical data without predefined rules. Additionally, ML-based trading can generate insights and support decision-making through algorithms like reinforcement learning and Q-learning [11].

In a study [12], researchers used reinforcement learning to optimize securities and stock exchanges. DeepMind mastered seven Atari video games, reaching human expert level in three, and eventually achieved expert-level performance in more than 20 different Atari games [13].

Machine learning (ML) has attracted considerable interest from scholars, which has led to the development of various algorithms for extracting information from different types of data. Initially, most ML algorithms were confined to prediction.

Machine learning algorithms, like deep reinforcement learning (DRL), are able to provide optimal decisions independently in complicated settings, such as financial markets. It has made ML a captivating study area in finance. DRL has shown significant achievement in solving complicated sequential decision-making problems. Research in Reinforcement Learning (RL) focuses on multi-agent frameworks, which are used in various fields. Fewer studies have examined multi-agent RL [14], especially in financial trading [15].

The Fractal Market Hypothesis (FMH), introduced by Peters, offers a substitute for the generally recognized efficient market theory. FMH suggests that market price behaviour is not random, as assumed in EMH, but instead exhibits a fractal characteristic with a comparable framework across various time intervals. It describes market behaviour with chaos, fractals, breakdowns, and crises. The Elliott wave theory [13] suggests that price has repetitive behavior across different time frames, with markets following repetitive cycles based on collective trading behavior.

Several studies suggest that the market's fractal nature may be attributed to traders' intuition, which involves varying data interpretations and investment horizons [15]. Financial markets exhibit unique characteristics within specific time frames. According to the Fractal Market Hypothesis (FMH), there may be interrelationships between different periods due to varying interpretations of information. The system is made using an LSTM network integrated with Q-learning, an RL algorithm that operates outside the policy to identify the optimal action based on the present conditions.

The Q-learning algorithm operates outside the current policy and seeks to maximize total reward. Our framework is built upon a deep Q network. Analysts have used various machine learning methods for stock showcase determination [15-19]. Deep learning models are gaining popularity in financial trading [20-22]. Ding et al. [23] presented a deep convolutional neural network for predicting stock prices.

In RL, the agent learns to link states with actions in order to maximize rewards [29]. The agent tracks state information from its environment and autonomously learns to choose actions based on the reward it receives and the current state [14].

The use of DRL strategies has obtained popularity in various trading methods [36-37] and has been assessed for their effectiveness in stock markets [38]. A suggested three-layer multi-group strategy outperformed a traditional buy-and-hold approach [39]. Our work employs this approach as a benchmark.

2 BACKGROUND OF THE RESEARCH

Touché and Safel [12] presented Repetitive Reinforcement Learning (RRL), a coordinated reinforcement method with a better performance compared to Q-learning. This RRL agent employs a single-layer neural network for maximizing a risk-adjusted return function based on previous returns and yield. Dorson et al. [31] employed RRL and Q-learning strategies to optimise investment strategies, integrating reinforcement learning with a trading system. They demonstrated a solid execution of the Q-learning.

In a specific data setup, the results outperformed the buy-and-hold strategy by a factor of two. Nomiwaka et al. [32] conducted an extensive experimental use of RL to solve the optimal execution problem by large-scale NASDAQ market microstructure data bases. These authors utilized historical records from INET and performed tests on three stocks - Qualcomm (QCOM), NVIDIA (NVDA), and Amazon (AMZN). The results demonstrated that RL outperformed the send-and-leave (S&L) strategy previously used in a basic market. Additionally, [33] applied adaptive reinforcement learning (ARL) in the financial market. They integrated a dynamic hyperparameter optimization layer and a risk management layer and examined the framework using two years of real EUR/USD data (January 2000 - January 2002), with 1-minute granularity, and achieved an average annualized return of 26%. Li et al. [34] presented a novel stock trading framework according to RL. The proposed system, MQ-trader, comprises four collaborative Q-learning agents.

Cumming et al. [35] used an LSTD-based RL algorithm (Reinforcement Learning Algorithm based on Least Squares Temporal Difference) to achieve an annual profit of 1.64% in the EUR/USD market. Akita et al. [24] found that combining textual and numerical data is more effective for predicting stock prices. [25] suggested an LSTM-based model by the use of technical indicators and historical price measures for anticipating stock price movements. [26] developed the ATT-ERNN model for anticipating stock price movements based on global events, using a two-stage attention-based recurrent neural network. Zhao et al. [28] used a market-aware framework to predict stock returns by combining fundamental and technical indicators. Caesar and Ozbayoglou [41] used technical analysis indicators and stock price data to label images for a CNN model.

In their research, Wen and Yuan [42] used a hybrid LSTM-CNN model to analyze data, incorporating various inputs such as Sunspot data, technical analysis indicators, fundamental analysis information, and economic indicators. According to their findings, the hybrid model showed better performance compared to individual CNN and LSTM models, particularly in terms of parameters such as Score-F1. Kim and Kim [43] also employed a combined CNN-LSTM model to predict S&P 500 share prices, outperforming individual LSTM and CNN methods. Additionally, Caesar et al. [44] utilized a different approach to predict buy, sell, and hold signals, concluding that their method showed higher effectiveness compared to the buy-and-hold approach. In their study, Ghorbani et al. [45] used a combined colored Petri nets strategy and genetic algorithm for simulating and forecasting buy/sell signals of stock exchanges. They found this method to be more effective in generating correct signals in comparison with other techniques, like neural networks, decision trees, and linear regression. Melki et al. [46] focused on efficiently summarizing and visualizing stock market information due to the large amount of data generated in the Tehran stock market. They used the incremental clustering method along with the k-means algorithm to identify stock movement signals. Chen et al. [47] concentrated on predicting stock trading signals, emphasizing the unstable and complex nature of stock market impacted by various interconnected variables.

In another study, Li et al. [48] discuss the challenges of algorithmic trading, proposing the TFJ-DRL model, which combines RL and deep learning for financial decision-making improvement.

Singh et al. (2022) discuss how the accessibility of information has transformed financial frameworks. They highlight how modern reinforcement learning can improve decision-making in complex financial scenarios. Gororge et al. [47] compared linear regression and SVM for stock price forecasting, aiming to demonstrate the advantages of SVM over linear regression.

3 RESEARCH METHODOLOGY

The descriptive and correlational research strategy aims to illustrate and understand existing conditions and relationships between factors using historical data. It examines cause and effect, identifies dependent and independent variables, and utilizes tools such as surveys, interviews, observations, tests, vouchers, and analysis for data collection.

This study uses a library strategy to collect specific data on the forex market from 2018 to 2021. Machine learning techniques in Python or MATLAB are applied for data analysis focusing on reinforcement learning (RL) to generate trading signals that maximize overall profit using the Q-learning algorithm in the stock market.

4 RESEARCH FRAMEWORK

An RL system comprises four major constituents: a value function, a reward signal, a policy, and, optionally, an environment model. The agent has interaction with the environment in discrete time steps, t = 0, 1, 2, 3, receiving information about the environment state at t time steps.

st ∈ S, where S denotes a set of possible states, and therefore an action is chosen.
An action, denoted as '∈ A(st)', belongs to the set of actions present in state 'st'.
After taking a step, the agent obtains a numerical reward, rt+1 ∈ ℝ, entering a new state, st+1.
The agent maps states to the probabilities of selecting different activities, known as the agent's approach (πt). The likelihood of choosing action a (at) when in state s (st) is represented by πt(s, a). An LSTM cell can retain and recall past data for an extended period. It includes an output gate, an input gate, and a forget gate, each of which holds values between 0 and 1. A value of 1 means to retain the data, and 0 means to discard it. The input gate determines which portions of the input data are added to the cell's content and to what degree. The forget gate makes a decision on which parts of the cell's content are removed. The output gate regulates which portions of the cell's content are included in the hidden state.

5 DISCOVERIES

The current work aims at predicting a specific exchange rate's medium-term trend by analyzing relationships among various exchange rates. We focused on cross-currency trades that reveal arbitrage opportunities in the forex market, exploiting short-term price differences between a currency and its associated currencies.
Triangular arbitrage involves three currencies, where one is derived from the exchange rates of the other two. This study focuses on the cross rates of GBP/USD, EUR/GBP, and EUR/USD, using EUR/USD as the base currency. In the foreign exchange market, the EUR/USD cross rate should be calculated based on the EUR/GBP and GBP/USD pairs. “Eq. (1)”:

Er=reur/usd=reue/gbp.rgbp/usd (1)

In connection (1), we mentioned the exchange rate as Er. To prevent arbitrage in regulated financial markets, a precise timing ratio for currency transactions is necessary. Short-term discrepancies in forex markets create trading opportunities for financial advisors. Analyzing features on a forex time series chart can reveal early differences in currency prices.
We analyze related currencies using triangular arbitrage to assess the medium-term direction of a currency. By using EUR/USD as the base currency alongside EUR/GBP and GBP/USD, we identify potential arbitrage opportunities. It is possible to apply this approach to other currencies with comparable monetary properties.
Profit function: To calculate efficiency and profit, if the price at time "t" is denoted as "pt", then the profit function is as follows, “Eq. (2)”:

Rit=log(pit/pit-1)*100 (2)

Function (2) is positive when the logarithm of the upper face of the number is 100.
Loss function, “Eq. (3)”:

$$CE = -\sum_{I=1}^{C=2} T_I \, LOG(S_I) = -t1\log(s1) - (1 - t1)\log(1 - s1) \quad (3)$$

In this scenario, there are two targets, labeled as 1C and 2C. The values (t0, t1) and S1 represent the actual label and output score of the network for target 1C, respectively.
Furthermore, 1 - t1 = 2t and 1 - S1 = 2S represent the actual label and output of the network for target 2C, respectively.

Inputs:
The dataset comprises values for the 2018-2021 period, with 1,325 data points during market hours. The EUR/USD ratio saw an increase of 680 and a decrease of 645 over this span.
Entry and exit signal:
The initial trading strategy involves using the stochastic oscillator.
This strategy involves using a version of stochastic volatility to generate buy and sell signals to enter and close trades.
Second trading strategy (mean reversion):
The average reversion strategy, which includes Bollinger Bands and directional averages, uses the directional average to identify the trend and Bollinger Bands to determine potential reversal points for entering transactions.
The implementation of the first strategy with the most optimal parameter values in each time step is carried out in the following sequence:
When there are no open transactions and the random oscillator exceeds 85, we enter a short position. The closing price is set at the last support level of the Donchian channels. To prevent losses, we use two stop loss levels: one based on the maximum allowed pips and the other on the Donchian channel position.
If there’s no open trade and the random oscillator is below 15, we will enter a long position. The trade will close at the latest resistance level in the 14-period Donchian channels. We will set two stop-loss values: one based on our maximum pip loss and the other on the Donchian channel position.
If the trade is open and its position is long, the trade will be closed if the value of the random oscillator is greater than 70. If the trade is open and its position is short, the trade will be closed if the value of the random oscillator is less than 30.
In the implementation of the second strategy, the following order is followed in each tick, using optimal parameters.

If no trade is open and the directional average exceeds 40 for the 14-day time frame, it indicates a strong trend. In that case, we will enter a short position if the closing price is beyond the upper Bollinger Band and the current price is less than or equal to that upper band for a longer time frame. A stop-loss will be set based on the maximum pips we are willing to risk.
If a trade is open with a long position and the current price value is lower than the middle band value of the Bollinger bands, the trade is closed.
If a trade is open with a short position and the current price value is above the middle band value of the Bollinger bands, the trade is closed.
We employed a deep learning algorithm to dynamically choose the values of four parameters for trading thresholds, loss limits, trading windows, and estimation windows. The values utilized as input for the deep learning algorithm are as follows:

Trading thresholds are set at a distance of 2.5 standard deviations on both sides of the average price difference, with an accuracy of 0.5 standard deviations.
The loss limit thresholds are set at a distance of four standard deviations on both sides of the average price difference, with an accuracy of 0.5 standard deviations.
The trading window is 15 minutes with a 10-minute accuracy.
The estimation window is 60 minutes with a 10-minute accuracy. The estimation window is 240 minutes with a 10-minute accuracy.

Turnover:
Here, data description, cleaning approaches, and feature extraction are discussed. The data includes the first trade price, the highest and lowest traded prices, the last trade price, and the trade volume, represented as candlestick charts (“Table” 1).

Table 1 This data is shown in the form of candlesticks

	Date	Time	Open Price	High Price	Low Price	Close Price	Volume
0	2000.01.02	23:15	1.0078	1.0087	1.0076	1.0086	41
1	2000.01.02	23:30	1.0087	1.0089	1.0079	1.0079	68
2	2000.01.02	23:45	1.0078	1.0132	1.0078	1.0129	66
3	2000.01.03	00:00	1.0129	1.0133	1.0120	1.0122	37
4	2000.01.03	00:15	1.0123	1.0125	1.0120	1.0124	37

The following sections explain each component depicted in the diagram of the presented model (“Fig. 1”).

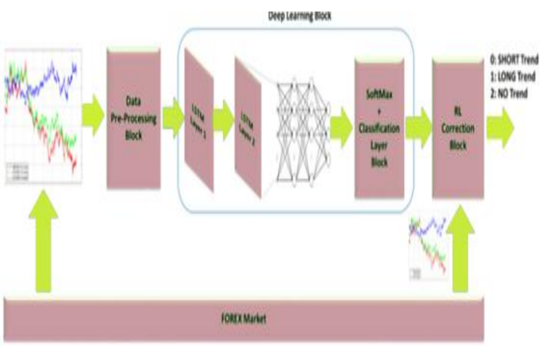

Fig. 1 Presented currency trend forecast model.

Data Preprocessing Block:
The text below explains the purpose of this part, which is to send financial period data. In this Block, the time series data is normalized within the range of 0 to 1, (“Table 2”).
To prepare the proposed model, the creator compiled a comprehensive database of financial data. They collected 99.9% accurate historical data for EUR/USD, GBP/USD, and EUR/GBP from 2018 to 2021. This included open, close, high, and low prices for each currency pair, excluding timestamps (CET time), to support the algorithm

The dataset has been classified into two sets: for training and for testing and validation. Training and validation simulations were conducted annually, as follows:
75% of the dataset was employed by training the model, whereas twenty-five percent was set aside for validation and testing. Both sets were analyzed to ensure equal representation of patterns (LONG, SHORT, INVALID) to prevent overfitting in the deep learning system.
Organized financial data is used as input for the preprocessing framework described in this section, and its output is processed by the subsequent deep learning component.

Table 2 Data pre-processing

AUDUSD	N/A	N/A	0.00345	0.53%
EURUSD	N/A	N/A	0.01184	1.10%
GBPUSD	N/A	N/A	0.01384	1.10%
NZDUSD	N/A	N/A	-0.0024	-0.39%
USDCAD	N/A	N/A	0.00768	0.57%
USDCHF	N/A	N/A	0.00251	0.29%
USDJPY	N/A	N/A	-0.853	-0.57%
EURCHF	N/A	N/A	0.01313	1.39%
EURGBP	N/A	N/A	-0.00148	-0.17%
EURJPY	N/A	N/A	0.823	0.51%
GBPJPY	N/A	N/A	0.951	0.50%
GBPCAD	N/A	N/A	0.02861	1.69%
CHFJPY	N/A	N/A	-1.796	-1.05%

CADCHF	N/A	N/A	-0.00081	-0.12%
NZDCAD	N/A	N/A	0.00021	0.03%
CADJPY	N/A	N/A	-1.227	-1.10%
AUDJPY	N/A	N/A	-0.058	-0.06%
EURCAD	N/A	N/A	0.02466	1.70%
EURAUD	N/A	N/A	0.00755	0.46%
EURNZD	N/A	N/A	0.02615	1.48%
GBPCHF	N/A	N/A	0.01611	1.45%
GBPAUD	N/A	N/A	0.00925	0.48%
GBPNZD	N/A	N/A	0.0299	1.45%
NZDCHF	N/A	N/A	5.0E-5	0.01%
NZDJPY	N/A	N/A	-0.953	-1.04%
AUDCAD	N/A	N/A	0.01047	1.19%
AUDCHF	N/A	N/A	0.0065	1.13%
AUDNZD	N/A	N/A	0.00993	0.93%
USDSGD	N/A	N/A	-0.00889	-0.66%
USDCNH	N/A	N/A	-0.01173	-0.16%
USDHKD	N/A	N/A	0.00352	0.05%
USDDKK	N/A	N/A	-0.0716	-1.03%
USDNOK	N/A	N/A	0.08572	0.81%
USDSEK	N/A	N/A	-0.0922	-0.88%
USDPLN	N/A	N/A	-0.0922	-2.29%
EURPLN	N/A	N/A	-0.05193	-1.20%
USDCZK	N/A	N/A	-0.4416	-1.88%
USDTRY	N/A	N/A	1.31109	4.27%
USDHUF	N/A	N/A	-0.5	-0.14%
USDZAR	N/A	N/A	-0.2017	-1.06%
USDMXN	N/A	N/A	-0.3342	-1.96%
EURZAR	N/A	N/A	0.0169	0.08%
USDRUB	N/A	N/A	0.0967	0.10%
EURRUB	N/A	N/A	N/A	N/A
USDILS	N/A	N/A	0.0504	1.39%
GBPTRY	N/A	N/A	2.04867	5.28%
EURSGD	N/A	N/A	0.00789	0.54%
EURHKD	N/A	N/A	0.08636	1.03%
EURTRY	N/A	N/A	1.74432	5.25%
GBPNOK	N/A	N/A	0.25732	1.94%
GBPDKK	N/A	N/A	-0.00761	-0.09%
NZDSGD	N/A	N/A	-0.0084	-1.02%
GBPSGD	N/A	N/A	0.00716	0.42%
GBPSEK	N/A	N/A	0.0205	0.16%
GBPZAR	N/A	N/A	0.012	0.05%
EURMXN	N/A	N/A	-0.1662	-0.91%
USDKRW	N/A	N/A	24.4	1.89%

Deep Learning Block:

This section provides forecasts for the short- and medium-term performance of the EUR/USD currency pair, focusing on Long Short-Term Memory (LSTM) networks. These networks are a kind of Recurrent Neural Network (RNN) that efficiently seize long-term dependencies. Each LSTM cell has three data ports—input, memory, and output—with a sigmoid activation function (σ) for processing. A tanh activation transforms the cell state and input. The cell retains values over time, while gates manage the flow of data in and out.

The mathematical model that describes the dynamics of a single LSTM cell is "Eq. (4)":

$$f_t = \sigma\left(W_f[h_{t-1.}x_t] + b_f\right).\, i_t = \sigma(W_i[h_{t-1.}x_t] + b_i).\; \tilde{C}_t = tanh(W_c[h_{t-1.}x_t] + b_c).\quad C_t = f_t * C_{t-1} + i_t * \tilde{C}_t,\; o_t = \sigma(W_o[h_{t-1.}x_t] + b_0).\, h_t = o_t * tanh(C_t), \tag{4}$$

- W_f, W_i, W_C, Wo present the LSTM weights
- b_f, b_i, b_C, bo denote the biases used for cells
- Ct denotes the state of the cell

In the LSTM network's data processing stage, input data from selected currencies is processed using specific equations to calculate the triangular spread. The network retains relevant information while discarding unnecessary data through a forget gate.

We used the LSTM architecture to classify input signals and aggregated data. Our deep learning model includes LSTM, fully linked layers, a classifier, and a SoftMax layer. The classifier assigns outputs to classes based on SoftMax probabilities and calculates loss and performance metrics.

Our model uses LSTM cells to process financial time series data, effectively classifying features via the classifier and SoftMax layer. It excels at feature extraction from LSTM, convolutional neural networks, and autoencoders, demonstrating strong detection capabilities.

The deep architecture that is proposed includes these components:

Input Layer:

LSTM layer (HiddenCellNumber);

LSTM layer (HiddenCellNumber);

Fully connected layer (NumberClasses)

Soft Max Layer:

Classification Layer:

In the developed model, the HiddenCellNumber variable representing the number of cells for each LSTM layer is 368, whereas NumberClasses is 3, i.e. (0): indicates a short process. (1) Shows the long trend. (2) No trend, because the currency is in a trading range, which is usually stated in financial terms to illustrate that a financial instrument does not present a clear trend.

In our model, each currency pair's input time series is segmented using a specific length as the input sequence for the deep learning model. Subsequent quotes form an input sequence to identify the economic trend of the target currency. In this case, we use 100 candles per input currency, representing one segment of the financial time series.

We are focusing on high-frequency trading (HFT) algorithms with 15-minute, 1-hour, and 4-hour time frames. Our analysis will use the closing prices for the EUR/USD currency pair, along with triangle arbitrage involving EUR/GBP, GBP/USD, and EUR/USD.

To make predictions, our deep learning system processes 100 closing price quotes for the mentioned currencies and generates an estimate for the trend of the next 100 quotes against the selected currency. This predictive estimate is used for trading, for instance, in the case of EUR/USD.

RL Correction Block:

This block aims to correct and verify the currency trends identified in the previous deep learning block. As shown by tests, the prior block's performance is consistent. For instance, when analyzing 2 million data points from 2021, its accuracy in predicting the EUR/USD trend averages 75%. The validation curve is in “Fig. 2”.

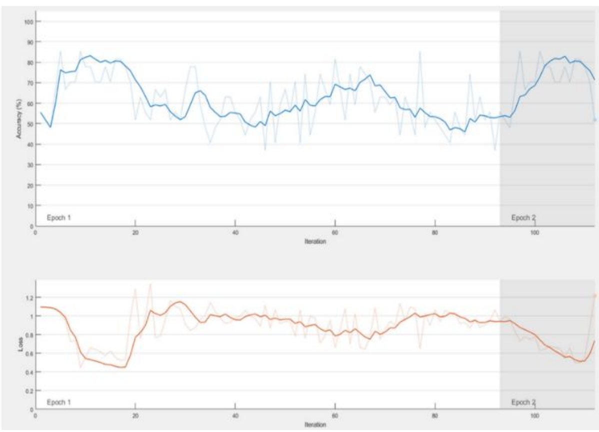

Fig. 2 Learning accuracy/validity of previous deep learning Block.

In this section, we aim to verify and correct the currency trends from the previous deep learning block, which has shown consistent performance. An analysis of 2 million instances from 2021 indicates that the block predicts the EUR/USD trend with an average accuracy of 75% during training, as shown in “Fig.2”. To enhance this process, the algorithm now includes an excessive learning layer utilizing unsupervised deep learning and RL methods.

This section outlines the RL (Reinforcement Learning) algorithm in “Fig. 3”, which uses two matrices: Wkin (x, y) for input synaptic weights and Wout (x, y) for output weights. Our study suggests a 900-cell RL motor map in a 30x30 grid, with initially randomly populated rows. Input weights might have real values, while output weights, Wout (xk, yk), can only be 0, 1, or 2, representing a short trend, long trend, and trading range, respectively.

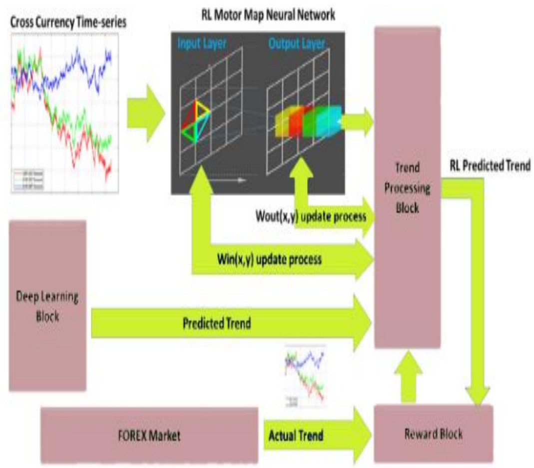

Fig. 3 Developed RL Block for correcting the expected trend made by the deep learning Block.

The learning process analyzes time series data for three currencies: GBP/USD, EUR/GBP, and EUR/USD. The normalized closing prices serve as input for the reinforcement learning (RL) engine. Each neuron in the input layer computes the distance between synaptic weights based on the EUR/USD forecast, resulting in a 30 x 30 distance matrix for the RL engine map: “Eq (5)”:

$$d(x.y) = \sqrt{\sum_{K=1}^{N}\left(w_{in}^{k}(x.y) - p_{c}^{\frac{EUR}{USD}}(T_K)\right) + \sum_{K=1}^{N}\left(W_{in}^{k}(x.y) - p_{c}^{\frac{GBP}{USD}}(tk)\right)^{2} + \forall x.y = 1..30; N = 100 +} \sum_{K=1}^{N}\left(w_{in}^{k}(x.y) - p_{c}^{\frac{EUR}{GBP}}(t_k)\right) \quad (5)$$

By using Equation (4) to obtain the distance matrix \(K(x, y) \), we identify the neuron that is closest to the input point. This selection is based on the Euclidean distance of the synaptic weights. In Poole's time series, we show the correlation between the neuron with the minimum distance \(K(x, y) \) and the coordinates \((xmin, ymin)\).

Let’s look at the position (xmin, ymin) in the RL motor map's output layer. This position produces a specific syntactic weight value that reflects the expected trend for the next 100 closing price quotes of the EUR/USD currency, allowing us to achieve practical results.

$$W_{out}(X_{min}.y_{min}) = \begin{cases} 0\ SHORT\ Trend \\ 1\ LONG\ Trend \\ 2\ TRADING\ range \end{cases} \quad (6)$$

At this stage, the RL system evaluates the system state, including Ytmin, Xtmin, DT(Xtmin, Ytmin), Wtin(xtmin, ytmin), $pEUR/USDc(tk)$, wtout(xtmin, ytmin), $pEUR/pEUR/US\ GBPc(tk)$, and $pGBP$/currency D.

The weights of the winning neuron at \((xtmin, ytmin)\) are adjusted to align the given currency with patterns of transition time segments. This training of the reinforcement learning engine helps it identify financial dynamics and corrects predictions from the previous deep learning block based on input data, (“Table 3”).

Table 3 Calculation of profit-loss

	Pred (no_act)	Pred (dec)	Pred (inc)
True (no_act)	-	False_dec_noact	False_inc_noact
True (dec)	-	True_dec	False_inc_dec
True (inc)	-	False_dec_inc	True_inc

	First 15 minutes	First 1 hour	first 4 hours
True (no_act)	73.9	68.31	79.42
True (dec)	64.24	63.91	68.58

By application of the labeling algorithm, a balanced distribution of three classes was achieved in the dataset. Labeling algorithm computes threshold values for time periods, leading to varying class distributions. “Table 4” shows the thresholds and the equivalent data points for each class in both the training and test sets.

Our models predict whether there will be an "increase" or "decrease" in the next 15 minutes, 1 hour, or 4 hours. When a prediction is made, a transaction occurs at the end of the test day for that time frame. If there’s no prediction, no transaction happens. A successful trade results in a profit if the prediction is accurate.

Table 4 Statistics of data sets (training and test sets)

Threshold	# of no_action	# of decrease	# of increase	
15 minutes	0.0023	412 (334–78)	400 (327–73)	402 (310–92)
at 1 o'clock	0.0040	413 (317–96)	414 (357–57)	385 (295–90)
4 hours	0.0055	400 (311–89)	422 (370–52)	388 (287–101)

Test on real data:

When analyzing time series data, the main objective is to predict the upcoming value. It's also feasible to forecast multiple future points by specifying the desired number of forecasts. However, during testing, the model's accuracy often diminishes with longer intervals. In 2017, Zhang et al. utilized a comparable model for predicting stock prices.

Our tests also covered 15-minute, 1-hour, and 4-hour forecasts of the directional movement of the EUR/USD currency pair.

We provide the total number of transactions for each test, along with accuracy results based on executed trades. The enhanced forecasting system integrates a deep learning module and a reinforcement learning (RL) framework, achieving an average trend forecasting accuracy of about 85%. “Table 5” displays validation results for the trend prediction line in “Fig. 3”, comparing outcomes with and without the RL-based trend corrections.

Table 5 Accuracy of trend forecasting performance for \\ EUR / USD currencies

Fiscal year	Deep learning prediction Block	Deep learning prediction Block + RL trend correction
2018	79.17 %	86.15 %
2019	67.12 %	85.29 %
2020	74.16 %	88.10 %
2021	75.41 %	85.65 %

It's evident from testing the proposed EUR/USD forecasting channel over several years that the trend prediction system built upon deep learning has lower accuracy compared to the modified RL line system.

Forex Trend Forecasting Program: HFT Network Trading System:

In previous chapters, we discussed predicting medium-term currency trends using triangular arbitrage and integrating deep learning and reinforcement learning. Now, it's crucial to develop a trading strategy based on these predictions. The system buys during forecasted LONG trends and sells during SHORT trends. An algorithm prevents opening positions too close together during portfolio fluctuations and aims to close existing positions during declines. The system does not enter trades when forecasts indicate a trading range. This trend prediction method has proven effective, especially for trading the EUR/USD crossover.

We tested our trading strategy using multi-currency data for GBP/USD, EUR/GBP, and EUR/USD from 2018 to 2021, achieving 99.9% accuracy. The simulations were conducted on an AMD Ryzen 16-core server with an Nvidia RTX 2080 Ti GPU and the full MATLAB toolbox. Our trading account started with USD, using a bid/ask size of 2 pips, a capital ratio of 1:400, and a specific LOT size for each FX trade.

The proposed method has been validated from 2018 to 2021 through experiments involving different daily market average windows. “Table 6” compares our method with similar ones in the literature, using the same forex currency for each comparison.

Table 6 Commercial performance analysis of the presented line

Method	FX Currency Cross	ROI (%)	MD (%)
Grid trading system	EUR/USD	95.18	12.25
Trading system	EUR/USD	62.79	23
Threshold (0.025%) – Strategy	EUR/USD	96.542	52.14
Threshold (0.075%) – Strategy	EUR/USD	65.145	8.90
Proposed	EUR/USD	98.23	15.97

The proposed High-Frequency Trading (HFT) method outperforms previous approaches by generating better returns and reducing loss-making operations. It operates on a 3-minute timeframe, executing 50 to 100 trades daily on the EUR/USD pair under favourable conditions. Analysis of "Table 6" reveals significant advantages in Return on Investment (ROI) compared to earlier versions and other methods. This HFT strategy has been effectively implemented in the currency market, offering valuable insights into performance when combined with average maximum drawdown (MD) data.
The ROI increased despite a slight decrease in comparison to the previous channel, justifying the small loss. The proposed algorithm's ROI is significantly lower than the suggested method, even with the same rate of return on invested capital. The method in the literature requires substantial capital investment under various conditions. Therefore, the decrease in Table 6 indicates that the designed profit return method outperforms the previously reported results by the authors in the EUR/USD crossover in the United States. "Table 7" also presents information about the changes in ROI and MD values over the test period of the proposed line.

Table 7 Planned line Business Performance Analysis Fan: Dynamic ROI / MD

Year(s)	FX Currency Cross	ROI Min (%)	MD Min (%)	ROI Max (%)	MD Max (%)
2018.2022	EUR/USD	62.79	8.90	98.23	52.14

In "Table 7", it is evident that the ROI remains almost unchanged within the validation sessions, while the capital utilization undergoes significant changes. It reaches maximum values to compensate for short-term losses caused by incorrect forecasting trends (e.g., MD index).
The data in "Table 6" confirms the effectiveness of the presented technique in anticipating medium-term trends using high-frequency trading (HFT) algorithms. Despite a slight decrease in performance compared to earlier versions, the method remains strong in forecasting trends, enabling better financial assessments and the development of trading strategies for a diverse portfolio to maximize gains and minimize losses.
We're adapting LSTM architectures and reinforcement learning frameworks for the STM32 architecture with integrated Cortex cores. Our goal is to create a portable financial platform, enabling users to monitor stocks, forecast financial trends, and conduct transactions using their mobile phones anytime and anywhere.
Time results of three forms: 15 minutes, 1 hour and 4 hours
Forecast for the next 15 minutes ("Table 8"):

Table 8 ME model: summary of the results of the next 15 minutes

Iterations=50	47.53	248/248
Iterations=100	56.00	45/248
Iterations=150	49.36	248/248
Iterations=200	65.25	59/248
Average	51.72	150.50/248

Technical model results ("Table 9"):

Table 9 TI model: summary of the results of the next 15 minutes

Iterations=50	**52.28**	**190/248**
Iterations=100	51.93	162/248
Iterations=150	54.11	179/248
Iterations=200	54.48	89/248
Average	52.19	158.24/248

Technical and fundamental results, ("Table 10"):

Table 10 ME_TI model: summary of the results of a future day

Iterations=50	48.62	235/248
Iterations=100	56.40	142/248
Iterations=150	47.65	189/248
Iterations=200	63.54	65/248
Average	54.12	158.12/248

Hybrid Model Results:
"Table 11" presents the number of transactions for each case and profit accuracy values in the model. In certain experiments with 200 iterations, our model produced very few transactions for both upward and downward predictions, with an average of 64.75 total predictions.

Table 11 Results of the combined model: 15-minute forecasts

Iterations		Hybrid model-modification based on ME_		Hybrid model-modification based on TI_	
ME	TI	Profit_accuracy (%)	# of transactions	Profit_accuracy (%)	# of transactions
50	50	70.80	137/248	70.80	137/248
50	100	73.50	117/248	74.36	117/248
50	150	69.60	125/248	77.60	125/248
50	200	81.63	52/248	82.35	52/248
100	50	78.13	32/248	65.63	32/248
100	100	69.23	26/248	65.38	26/248
100	150	70.59	34/248	70.59	34/248
100	200	73.17	46/248	75.00	46/248
150	50	76.56	128/248	78.13	128/248
150	100	72.64	106/248	78.30	106/248
150	150	73.17	123/248	80.49	123/248
150	200	100.00	8/248	100.00	8/248
200	50	80.00	20/248	76.47	20/248
200	100	80.77	28/248	80.77	28/248
200	150	84.00	26/248	82.61	26/248
200	200	83.33	28/248	85.71	28/248
Average		77.32	64.75/248	77.76	64.75/248

Forecast for the next 1 hour:
Results of fundamental model (“Table 12”):

Table 12 ME model: summary of the results of the next 1 hour

Iterations=50	57.44	242/248
Iterations=100	54.40	182/248
Iterations=150	39.83	236/248
Iterations=200	53.57	38/248
Average	51.31	174.50/248

Results of the technical model, (“Table 13”):

Table 13 TI model: Summary of the results of the next 1 hour

Iterations=50	43.31	157/248
Iterations=100	47.78	180/248
Iterations=150	51.37	146/248
Iterations=200	51.85	103/248
Average	48.58	146.50/248

Technical and macroeconomic model results, (“Table 14”):

Table 14 ME_TI model: summary of the results for the next 1 hour

Iterations=50	43.16	234/248
Iterations=100	43.81	226/248
Iterations=150	42.68	164/248
Iterations=200	85.71	10/248
Average	53.84	158.50/248

Results of hybrid model, (“Table 15”):

Table 15 The combined model: forecasts for the next 1 hour

ME	TI	profit_accuracy (%)	# of transactions	profit_accuracy (%)	# of transactions
50	50	58.39	137/248	57.66	137/248
50	100	58.71	155/248	56.13	155/248
50	150	61.60	125/248	61.60	125/248
50	200	94.74	21/248	95.00	21/248
100	50	76.06	71/248	71.83	71/248
100	100	69.41	85/248	70.59	85/248
100	150	79.37	63/248	80.95	63/248
100	200	100.00	2/248	100.00	2/248
150	50	67.44	86/248	70.93	86/248
150	100	70.80	113/248	74.34	113/248
150	150	69.79	96/248	73.96	96/248
150	200	95.00	43/248	95.00	43/248
200	50	77.78	9/248	75.00	9/248
200	100	84.62	13/248	84.62	13/248
200	150	100.00	9/248	100.00	9/248
200	200	100.00	14/248	100.00	14/248
Average		78.98	65.13/248	79.23	65.13/248

Forecast for the next 4 hours:
Results of macroeconomic model, (“Table 16”):

Table 16 ME model: summary of the results of the next 4 hours

Iterations=50	43.40	235/242
Iterations=100	47.11	242/242
Iterations=150	44.74	228/242
Iterations=200	54.00	120/242
Average	47.31	206.25/242

Technical model results, (“Table 17”):

Table 17 TI model: summary of the results of the next 4 hours

Iterations=50	48.13	187/242
Iterations=100	41.48	176/242
Iterations=150	45.73	164/242
Iterations=200	64.18	79/242
Average	49.88	151.50/242

Technical and macroeconomic model results, (“Table 18”):

Table 18 ME_TI model: summary of the results for the next 4 hours

Iterations=50	44.44	81/242
Iterations=100	42.72	206/242
Iterations=150	46.51	172/242
Iterations=200	61.25	96/242
Average	48.73	138.75/242

Hybrid model results, ("Table 19"):

Table 19 The combined model: forecasts for the next 4 hours

Iterations		Hybrid model-modification based on ME_		Hybrid model-modification based on TI_	
ME	TI	Profit_accuracy (%)	# of transactions	Profit_accuracy (%)	# of transactions
50	50	77.14	105/242	77.14	105/242
50	100	82.98	94/242	76.60	94/242
50	150	78.49	93/242	80.65	93/242
50	200	88.57	36/242	87.88	36/242
100	50	79.46	112/242	82.14	112/242
100	100	80.81	99/242	79.80	99/242
100	150	77.66	94/242	81.91	94/242
100	200	100.00	9/242	100.00	9/242
150	50	78.30	106/242	77.36	106/242
150	100	82.98	94/242	75.53	94/242
150	150	79.78	89/242	80.90	89/242
150	200	Nan	0/242	Nan	0/242
200	50	92.68	43/242	92.68	43/242
200	100	86.36	44/242	86.05	44/242
200	150	90.00	40/242	87.88	40/242
200	200	86.00	51/242	85.11	51/242
Average		84.08	69.31/242	83.44	69.31/242

Validation:

To validate our findings, we expanded our dataset to include a recent case, resulting in 1,687 data points with 785 increases and 845 decreases. We used a labeling algorithm to ensure a balanced distribution across three classes. The dataset statistics are given in "Table 20".

Table 20 Statistics of extensive datasets (training and testing sets)

Threshold	# of no_action	# of decrease	# of increase	
15 minutes ahead	0.0022	497 (438–59)	515 (464–51)	507 (465–42)
An hour ahead	0.0040	507 (451–56)	527 (476–51)	483 (438–45)
4 hours ahead	0.0054	503 (448–55)	532 (483–49)	480 (432–48)

The dataset was split into 10% for testing and 90% for training. The prediction results for all three forms of the proposed hybrid model, according to the developed data, are reported below.

Forecast 15 minutes, ("Table 21"):

Table 21 Combined model (on extended data set): 15 min forecasts

Iterations		Hybrid model-modification based on ME_		Hybrid model-modification based on TI_	
ME	TI	Profit_accuracy (%)	# of transactions	Profit_accuracy (%)	# of transactions
50	50	55.42	83/152	53.01	83/152
50	100	59.38	96/152	61.46	96/152
50	150	74.63	67/152	76.12	67/152
50	200	81.82	52/152	81.82	52/152
100	50	64.18	67/152	67.16	67/152
100	100	59.49	79/152	65.82	79/152
100	150	66.04	53/152	73.58	53/152
100	200	75.34	84/152	74.65	84/152
150	50	60.47	86/152	55.81	86/152
150	100	57.73	97/152	61.86	97/152
150	150	69.12	68/152	75.00	68/152
150	200	84.13	75/152	84.38	75/152
200	50	83.08	71/152	81.36	71/152
200	100	79.31	67/152	79.66	67/152
200	150	79.31	64/152	78.18	64/152
200	200	85.45	62/152	85.19	62/152
Average		70.93	73.19/152	72.19	73.19/152

Forecast 1 hour, ("Table 22") and 4 hours forecast, ("Table 23"):

Table 22 Combined model (on extended dataset): 1-hour forecasts

Iterations		Hybrid model-modification based on ME_		Hybrid model-modification based on TI_	
ME	TI	Profit_accuracy (%)	# of transactions	Profit_accuracy (%)	# of transactions
50	50	60.71	56/152	57.14	56/152
50	100	71.43	42/152	66.67	42/152
50	150	51.79	56/152	58.93	56/152
50	200	100.00	2/152	100.00	2/152
100	50	63.86	83/152	53.01	83/152
100	100	73.02	63/152	61.90	63/152
100	150	58.11	74/152	58.11	74/152
100	200	86.67	55/152	88.68	55/152
150	50	82.00	50/152	74.00	50/152
150	100	65.00	40/152	67.50	40/152
150	150	62.50	48/152	64.58	48/152
150	200	81.25	37/152	82.35	37/152
200	50	88.89	10/152	88.89	10/152
200	100	83.33	6/152	83.33	6/152
200	150	57.14	8/152	62.50	8/152
200	200	62.50	8/152	57.14	8/152
Average		71.76	39.88/152	70.30	39.88/152

Table 23 Combined model (on the extended dataset): 4-hour forecasts

Iterations		Hybrid model-modification based on ME_		Hybrid model-modification based on TI_	
ME	TI	Profit_accuracy (%)	# of transactions	Profit_accuracy (%)	# of transactions
50	50	79.66	59/152	71.19	59/152
50	100	67.21	61/152	67.21	61/152
50	150	68.85	61/152	60.66	61/152
50	200	75.34	84/152	72.73	84/152
100	50	77.61	67/152	73.13	67/152
100	100	62.86	70/152	61.43	70/152
100	150	67.14	70/152	62.86	70/152
100	200	75.00	77/152	71.88	77/152
150	50	79.69	64/152	75.00	64/152
150	100	69.12	68/152	67.65	68/152
150	150	63.77	69/152	59.42	69/152
150	200	75.68	84/152	72.73	84/152
200	50	71.64	71/152	72.46	71/152
200	100	66.67	84/152	66.22	84/152
200	150	67.61	81/152	68.06	81/152
200	200	72.06	75/152	69.35	75/152
Average		71.24	71.56/152	68.25	71.56/152

6 CONCLUSIONS

The proposed hybrid model underwent testing with recent datasets. The test results were consistent with previous findings, presenting just a slight decline in accuracy gains. This work shows that it is possible to use technical and macroeconomic indexes separately or together for training LSTM models for predicting currency pair movements in Forex. Processing these indicators with separate LSTMs and integrating the results through intelligent decision logic enhances prediction accuracy. A third category was added to capture small price changes between consecutive days, improving accuracy in direction prediction. The study introduces a method for defining the unchanged class threshold. LSTM networks were trained for predicting the next day's value and values of k days onward, although they slightly decreased accuracy when predicting three periods. The research also shows that increasing training iterations boosts accuracy, along with reducing the transaction numbers, which consequently lowers potential profit and risk. Future research will extend this approach to include additional currency pairs like EUR/GBP, GBP/USD, and others. We propose developing a trading simulator to validate the model and observe its real-time behaviour, while addressing key trading issues such as account management and leverage ratios. Additionally, we plan to explore the use of deep learning methods for structuring financial portfolios with diverse instruments, allowing a reinforcement learning agent to optimize allocations to minimize risk and maximize profits.

REFERENCES

[1] Benita, F., López-Ramos, F., Nasini, S. A., Bi-Level Programming Approach for Global Investment Strategies with Financial Intermediation, Eur. J. Oper. Res., Vol. 274, 2019, pp. 375–390.

[2] Liu, Z., Wang, J., Supply Chain Network Equilibrium with Strategic Financial Hedging Using Futures, Eur. J. Oper. Res.,Vol. 272, 2019, pp. 962–978.

[3] Sermpinis, G., Stasinakis, C., Rosillo, R., and De La Fuente, D., European Exchange Trading Funds Trading with Locally Weighted Support Vector Regression. Eur. J. Oper. Res., Vol. 258, 2017, pp. 372–384.

[4] Doyle, J. R., Chen, C. H., Patterns in Stock Market Move Ments Tested as Random Number Generators, Eur. J. Oper. Res., Vol. 227, 2013, pp. 122–132.

[5] Oztekin, A., Kizilaslan, R., Freund, S., and Iseri, A., A Data Analytic Approach to Forecasting Daily Stock Returns in an Emerging Market. Eur. J. Oper. Res., Vol. 253, 2016, pp. 697–710.

[6] Zhang, J., Cui, S., Xu, Y., Li, Q., and Li, T., A Novel Data-Driven Stock Price Trend Prediction System, Expert Syst. Appl., Vol. 97, 2018, pp. 60–69.

[7] Chou, J. S., Nguyen, T. K., Forward Forecast of Stock Price Using Sliding-Window Metaheuristic-Optimized Machine-Learning Regression, IEEE Trans. Ind. Inform., Vol. 14, 2018, pp. 3132–3142.

[8] Delaney, L., Investment in High-Frequency Trading Technology: A Real Options Approach, Eur. J. Oper. Res., Vol. 270, 2018, pp. 375–385.

[9] Fischer, T., Krauss, C., Deep Learning with Long Short-Term Memory Networks for Financial Market Predictions, Eur. J. Oper. Res., Vol. 270, 2018, pp. 654–669.

[10] Long, W., Lu, Z., and Cui, L., Deep Learning-Based Feature Engineering for Stock Price Movement Prediction. Knowl. Based Syst., Vol. 164, 2019, pp. 163–173.

[11] Sutton, R. S., Learning to Predict by The Methods of Temporal Differences, Mach. Learn., Vol. 3, 1988, 3, pp. 9–44.

[12] Moody, J., Saffell, M., Learning to Trade Via Direct Reinforcement, IEEE Trans. Neural Netw., Vol. 12, 2001, pp.875–889.

[13] Sutton, R. S., Temporal Credit Assignment in Reinforcement Learning. Ph.D. Thesis, University of Massachusetts Amherst, Amherst, MA, USA, 1985.

[14] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M., Playing Atari with Deep Reinforcement Learning. ArXiv 2013, arXiv: 1312.5602.

[15] Chung, H., Shin, K. S., Genetic Algorithm-Optimized Long Short-Term Memory Network for Stock Market Prediction, Sustainability, Vol. 10, 2018, 3765.

[16] Carta, S., Corriga, A., Ferreira, A., Recupero, D. R., and Saia, R., A Holistic Auto-Configurable Ensemble Machine Learning Strategy for Financial Trading, Computation, Vol. 7, 2019, pp. 67.

[17] Carta, S., Medda, A., Pili, A., Reforgiato, D. R., Saia, R., Forecasting E-Commerce Products Prices by Combining an Autoregressive Integrated Moving Average (ARIMA) Model and Google Trends Data, Future Internet, Vol. 11, 2019, pp. 5.

[18] Vukovic, D., Vyklyuk, Y., Matsiuk, N., and Maiti, M., Neural Network Forecasting in Prediction Sharpe Ratio: Evidence from EU Debt Market. Phys. A Stat. Mech. Appl., Vol. 542, 2020, pp. 123331.

[19] Maiti, M., Vyklyuk, Y., and Vuković, D., Cryptocurrencies Chaotic Co-Movement Forecasting with Neural Networks. Internet Technol. Lett., Vol. 3, 2020, pp. 157.

[20] Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., and Salwana, E., Deep Learning for Stock Market Prediction, Entropy, Vol. 22, 2020, pp. 840.

[21] Nabipour, M., Nayyeri, P., Jabani, H., Shahab, S., and Mosavi, A., Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; A Comparative Analysis, IEEE Access, Vol. 8, 2020, pp. 150199–150212.

[22] Le Cun, Y., Bengio, Y., and Hinton, G. J., Hinton. Deep. Learn., Vol. 521, 2015, pp. 436.

[23] Ding, X., Zhang, Y., Liu, T., and Duan, J., Deep Learning for Event-Driven Stock Prediction, In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, Vol. 25–31, 2015, pp. 2327–2333.

[24] Akita, R., Yoshihara, A., Matsubara, and T., Uehara, K., Deep Learning for Stock Prediction Using Numerical and Textual Information, In Proceedings of the 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, Vol. 26–29, 2016, pp. 1–6.

[25] Nelson, D. M., Pereira, A. C., De Oliveira, R. A., Stock Market's Price Movement Prediction with LSTM Neural Networks, In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, Vol. 14–19, 2017, pp. 1419–1426.

[26] Liu, J., Chen, Y., Liu, K., and Zhao, J., Attention-Based Event Relevance Model for Stock Price Movement Prediction, In Communications in Computer and Information Science, Proceedings of the China Conference on Knowledge Graph and Semantic Computing, Chengdu, China, Vol. 26–29, 2017, pp. 37–49.

[27] Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., and Cottrell, G., A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction, 2017, pp. 1704.02971.

[28] Zhao, R., Deng, Y., Dredze, M., Verma, A., Rosenberg, D., and Stent, A., Visual Attention Model for Cross-sectional Stock Return Prediction and End-to-End Multimodal Market Representation Learning, In Proceedings of the Thirty-Second International Flairs Conference, Sarasota, FL, USA, Vol. 19–22, 2019.

[29] Sutton, R. S., Barto, A. G., Introduction to Reinforcement Learning; MIT Press: Cambridge, MA, USA, 1998.

[30] Gold, C., FX Trading Via Recurrent Reinforcement Learning, In Proceedings of the 2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003, Proceedings, Hong Kong, China, Vol. 20–23, 2003, pp. 363–370.

[31] Duerson, S., Khan, F., Kovalev, V., and Malik, A. H., Reinforcement Learning in Online Stock Trading Systems, Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.83.5299&rep=rep1&type=pdf, 2021.

[32] Nevmyvaka, Y., Feng, Y., and Kearns, M., Reinforcement Learning for Optimized Trade Execution, In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, Vol. 25–29, 2006, pp. 673–680.

[33] STMicroelectronics S.r.l.–ADG Central R & D Group, 95121 Catania, Italy Dempster, M. A., Leemans, V., An Automated FX Trading System Using Adaptive Reinforcement Learning, Expert Syst. Appl., Vol.30, 2006, pp. 543–552.

[34] Lee, J. W., Park, J., Jangmin, O., Lee, J., and Hong, E., A Multiagent Approach to $ Q $-Learning for Daily Stock Trading. IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum., Vol. 37, 2007, pp. 864–877.

[35] Cumming, J., Alrajeh, D. D., and Dickens, L., An Investigation into the Use of Reinforcement Learning Techniques within the Algorithmic Trading Domain, Master's Thesis, Imperial College London, London, UK, 2015.

[36] Xiong, Z., Liu, X. Y., Zhong, S., Yang, H., and Walid, A., Practical Deep Reinforcement Learning Approach for Stock Trading, 2018, pp. 1811.07522.

[37] Wu, X., Chen, H., Wang, J., Torino, L., Lia, V., and Fujita, H., Adaptive Stock Trading Strategies with Deep Reinforcement Learning Methods, Inf. Sci., Vol. 538, 2020, pp. 142–158.

[38] Carta, S., Corriga, A., Ferreira, A., Podded, A. S., and Recupero, D. R., A Multi-Layer and Multi-Ensemble Stock Trader Using Deep Learning and Deep Reinforcement Learning, Appl. Intel., Vol. 51, 2021, pp.889–905.

[39] Carta, S., Ferreira, A., Podded, A. S., Recupero, D. R., and Sana, A., Multi-DQN: An Ensemble of Deep Q-Learning Agents for Stock Market Forecasting. Expert Syst. Appl., Vol. 164, 2021, pp. 113820.

[40] Seer, O. B., Ozbayoglu, A. M., Financial Trading Model with Stock Bar Chart Image Time Series with Deep Convolutional Neural Networks, 2019, pp. 1903.04610.

[41] Wen, Y., Yuan, B., Use CNN-LSTM Network to Analyze Secondary Market Data, In Proceedings of the 2nd International Conference on Innovation in Artificial Intelligence, 2018, pp. 54-58.

[42] Kim, T., Kim, H. Y., Forecasting Stock Prices with a Feature Fusion LSTM-CNN Model Using Different Representations of the Same Data, Plops one, Vol. 14, No. 2, 2019, pp. e0212320.

[43] Seer, O. B., Ozbayoglu, A. M., Algorithmic Financial Trading with Deep Convolutional Neural Networks: Time Series to Image Conversion Approach, Applied Soft Computing, Vol. 70, 2018, pp. 525-538

[44] Ghorbanifar, Yahyazadeh Far, M., Nabavi Chashmi, Stock Trading Signal Prediction Using Colored Petri Nets and Genetic Algorithm (Case Study: Tehran Stock Exchange), Research Journal of Executive Management, Vol. 11, No. 21, 2019, pp.205-227.

[45] Maleki Moghadam, P. A., Alam Tabriz, P., and Najafi., Designing an Intelligent Model to Determine Stock Trading Signals with A Data Mining Approach, New Researches in Mathematics, Vol. 6, No. 24, 2020, pp. 159-172.

[46] Chen, Y., Hao, Y., Integrating Principal Component Analysis and Weighted Support Vector Machine for Stock Trading Signals Prediction, Neurocomputing, Vol. 321, 2018, pp. 381-402.

[47] Lei, K., Zhang, B., Li, Y., Yang, M., and Shen, Y., Time-Driven Feature-Aware Jointly Deep Reinforcement Learning for Financial Signal Representation and Algorithmic Trading, Expert Systems with Applications, Vol. 140, 2020, pp. 112872.

[48] An, B., Sun, S., and Wang, R., Deep Reinforcement Learning for Quantitative Trading: Challenges and Opportunities, IEEE Intelligent Systems, Vol. 37, No. 2, 2022, pp. 23-26.

[49] Singh, V., Chen, S. S., Singhania, M., Nanavati, B., and Gupta, A., How Are Reinforcement Learning and Deep Learning Algorithms Used for Big Data Based Decision Making in Financial Industries–A Review and Research Agenda, International Journal of Information Management Data Insights, Vol. 2, No. 2, 2022, pp. 100094.

[50] Brim, A., Flann, N. S., Deep Reinforcement Learning Stock Market Trading, Utilizing a CNN with Candlestick Images, Plos one, Vol. 17, No. 2, 2022, pp. e0263181.

[51] Cheng, L. C., Huang, Y. H., Hsieh, M. H., and Wu, M. E., A Novel Trading Strategy Framework Based on Reinforcement Deep Learning for Financial Market Predictions, Mathematics, Vol. 9, No. 23, 2021, pp. 3094.