Deep Reinforcement Learning of the Rules of Financial Assets Trading

Soroush Barmaki[1]*, Morteza Zahedi[2]

[1]*Postgraduate Student of Department of Artificial Intelligence, Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran.*

[2]*Assistant Professor of Department of Artificial Intelligence, Faculty of Computer Engineering, Shahrood University of Technology, Shahrood, Iran.*

**Corresponding author*

Abstract

The foreign exchange (Forex) market is the world's largest currency exchange market where diverse people globally make financial investments. Since the advent of this market, a leading challenge for capital investors and analysts has been devising a versatile solution and strategy to make profitable investments in this market. Recently, machine learning (ME) and, particularly, deep learning (DL) algorithms, in tandem with intelligent learning systems working based on previous data, have markedly fostered functions such as data classification or feature extraction. Likewise, the advent of reinforcement learning (RL) and intelligent systems has revolutionized learning and operating with no need for previous data and merely based on algorithms such as Q-Learning. The present research operated an intelligent agent (IA) trained based on DL and Q-Learning algorithms in RL to make financial transactions in the Forex market that deliver the greatest profitability and least losses within the test period. For several major currencies in the Forex market, this RL-based IA was compared for 24 hours with a trading strategy based on trend detection algorithms employing diverse ME classification techniques. The proposed IA delivered higher profitability values of 64.9% and 38.5% compared to the previous structures in aud/usd and eur/usd currencies, respectively. Collectively, the IA delivers higher profitability than the ME-based trading strategies.

Keywords: Machine learning, Deep learning, Deep reinforcement learning, Forex market, Trading strategy

1. Introduction

The foreign exchange (Forex) market is noticeably lucrative for investment and capital gain. Such a remarkable potency inspires diverse investors worldwide to make profitable investments in this market. However, the Forex market does not ensure the profitability of all transactions, and investors to achieve profit and income require a deep insight into this market and its governing rules. This entails extensive research to assess the efficacy of the market's key determinants and explore a tailored and reliable solution for performing relative transactions. The Forex market is exceedingly chaotic, and the best strategy to trade in this market is to deliberately move parallel with its underlying patterns and governing trends. For this to be realized, one strategy is to employ a reinforcement learning (RL)-based intelligent agent (IA) that decides about currency pairs while reflecting the conditions governing the Forex market.

Wang et al. (1) devised an end-to-end deep Q-trading system to take a reasonable position at each trading time step. Xiong et al. (2) employed the deep deterministic policy gradient (DDPG) algorithm to learn a dynamic stock trading strategy. They claimed that the DDPG algorithm outperformed both the Dow Jones Industrial Average (DJIA) and the traditional min-variance portfolio allocation strategy, with a profit that was roughly twice that of the minimum-variance strategy.

Li et al. (3) evaluated the performance of three models in deep reinforcement learning (DRL), including classic Deep Q-Network (DQN), Double DQN (DDQN), and Dueling DDQN in learning trading strategy for ten US stocks. The rationale for their study was a report by Van et al. (4) who proved that the error of overestimation of the q-value in DQN can intensify errors of the DQN model in a spiral course. Regarding the above three models, Li et al. found that the classic DQN outperforms the two other models in making financial decisions.

Luo et al. (5) merged two convolutional neural networks (CNN) (as feature extractors) with a DDPG model to learn trading strategies on real-time stock indices futures. They used these two CNNs to implement an actor-critic RL algorithm in the Chinese market and optimize them employing TD Error and Policy Gradient, thereby achieving marked profit within the test period.

Zarkias et al. (6) proposed a novel price trailing technique that reformulates trading as a control problem, thereby presenting trading strategies that allow for capturing the price trends and making profitable decisions. The proposed technique was based on following the market trend and taking a sustained position in the financial market. The model they adopted was that presented by Van et al. (4), which has a DRL structure with two deep reinforcement agents.

Zhang et al. (7) adopted DRL algorithms to design trading strategies for continuous futures contracts. They evaluated discrete and continuous action environments employing a reward function with volatility scaling. By running DRL algorithms (i.e., DQN, Policy Gradient, and Actor-Critic), they found that the DQN algorithm offers higher profitability than other algorithms in nearly all transactions. They concluded that upon using a linear efficiency function, the RL algorithm and the modern portfolio theory will be equivalent.

Theate et al. (8) incorporated the Sharpe ratio performance indicator (as the key index for evaluation) in their proposed model. Then, by running the TDQN algorithm, they substituted the initial CNN part of the classic DQN with DNN. They further used open, high, low, and close (OHLC) prices and trading volume in the state environment. Their proposed model outperformed buy-and-hold, sell-and-hold, and two trend detection methods regarding profitability.

Wu et al. (9) worked with the Gated Recurrent Unit (GRU) to extract temporal dependencies from raw financial data and technical indicators. They further worked with DQN and Deterministic Policy Gradient (DPG) algorithms to learn single-stock trading strategies. In this method, mini_batch first serves as an input to the GRU-based Actor network. Next, the selected Action and the current standing of the actor are considered to determine the q-value in the MLP network. Contrary to classic DRL algorithms that only use the Actor section, their proposed method incorporates the Critic section into this structure. By comparison, the proposed method outperforms the classic DQN structure by offering higher profits and lower losses.

Suchaimanacharoen et al. (10) worked with CNN to predict the future prices of the eur/usd currency pair. Next, they fed the output (i.e., the predicted prices) to the Policy Gradient (PG) model to learn the trading strategy within the high-frequency trading (HFT) algorithm. By conducting experiments on 30-minute intervals of eur/usd pair in Forex, they incorporated the closed price to the CNN (for 240 periods of data, equivalent to 5 days, as well as the next 24 periods, equivalent to 12 hours). They further incorporated the predicted closed price to a DNN while merging these 24 prices with other features, passing them through an MLP network, and eventually training the reinforcement factor. By these, they successfully made the right diagnosis of long and short positions in the Forex market and attained nearly 40% profit in the test period.

Lee et al. (11) proposed a time-driven feature-aware jointly DRL model (TFJ-DRL) that can learn feature representation from highly nonstationary and noisy financial time series and simultaneously extract temporal dependencies. This model creates a weighted feature vector by extracting OHLC prices, volume of transactions, related currencies, and technical analysis. Then, by passing through a GRU and Attention-based algorithm, the model displays primary features. As such, the reinforcement agent possesses its best performance and optimizes its policy based on price changes, all based on this display and the previous action. Their proposed model outperformed other models in all cases regarding profitability.

Although diverse time series models are used based on historical data for a better representation of price movements, there exist some unresolved issues, with the proposed models failing to learn a tailored qualitative feature vector that reflects the behavior of past price movements. A versatile way of financial representation to illustrate transient price movement behaviors is the Japanese Candlestick Charting (JCC) technique. JCC is a well-established technique of showing price fluctuations over time and was first devised by the Japanese trader Manhisa Homma to predict variations in the price of rice (12).

Hu et al. (13) proposed a novel investment decision strategy (IDS) based on Convolutional autoencoders (CAEs) from the JCC stock representation techniques. They made profitable transactions by using the CAEs (14) to create features of stock prices and then grouping these features and building a portfolio based on the created groups.

Thammakesorn et al. (15) proposed a model to generate stock trading strategies using features that effectively recognize good patterns in candlestick charts. This model builds the feature vector required to develop a machine learning (ME) model by extracting various features based on the candlestick charts between 1 and 3 days. Again, based on the model proposed in (16), they developed a decision tree to choose between buying, selling, or no transaction based on these candlestick features. They reported that the proposed model is more profitable than the conventional methods based on technical analysis.

Orquin et al. (17) tested the efficiency of the eur/usd pair by simply reflecting the candlestick charts only for the price itself. They found that an hourly timeframe for this currency pair is more profitable according to the implemented strategy.

Birogul et al. (18) employed the object detection system YOLO (You Only Look Once) to recognize patterns in candlestick charts to generate buy/sell signals for a stock. They principally aimed at tracking the right signals in the market and making correct predictions. They achieved an insignificant profit for individual stocks and found that a measurable profit is attainable by adding small profits on different currencies. Indeed, their research is more of a portfolio management instead of a trading strategy for a single stock.

Fengqian et al. (19) proposed an innovative model to generate trading strategies on single stocks using candlestick charts. They proposed an RL technique to learn stock trading time based on the current pattern recognized from candlestick price charts at each time step. Pattern recognition was performed by clustering similar patterns in candlestick charts using the K-means algorithm.

The present research creates an IA based on diverse approaches and various features of the Forex market. The aim is to assess these features for realizing long-term profit and capital acquisition through trading.

Importantly, by setting the optimal policy for this IA and making decisions based on the market situation, the proposed IA can be directly used for profitable transactions. Thus, the traders will no longer need to reflect a multitude of current analyses. This IA can further serve as a smart advisor. The traders need to simply use this agent to confirm or reject their operational decisions in the market, besides their fundamental and technical analyses.

The contribution of this research is building multiple novel structures to process the IA state and make relevant decisions. Elsewhere, Taghian et al. (20) have proposed several structures to extract features from financial markets. The present research sequentially merges beneficial structures of the models proposed in (20) to perform feature extraction. Evidently, the proposed model works sounder for both aud/usd and eur/usd currencies than the previously proposed structures and is more profitable for both gbp/usd and usd/cad currencies.

2. Research methodology

2.1. Data collection

Currency pairs aud/usd, eur/usd, gbp/usd, and usd/cad were selected based on a daily period. The research data were gathered from 01 January 2001 to 01 January 2023. Training data were used from 01 January 2001 to 10 January 2019. Likewise, validation data were used from 01 February 2019 to 10 January 2020, and evaluation and test data were employed from 10 February 2020 to 01 January 2023.

2.2. Machine learning models for trend detection

This research investigated Decision Tree, Gradient Boosting, Gaussian Naïve Bayes, K-nearest neighbor, Logistic Regression, Multilayer Perceptron (MLP), Random Forest, and Support Vector Machine (SVM) models. For training these models, the current market trend was first recognized based on the market situation and the closed price for a certain number of days in the past. Then, some features were extracted from the raw data set of OHLC prices, followed by training the models based on the extracted features and trend labeled as the classification objective. Notably, model training needs to cover and investigate the impacts of diverse parameters.

2.3. DRL-based IAs

Extracting diverse features from the same data makes more data available and delivers the most robust outputs. In this research, various states were made up of diverse features of the input data. These include the following.

2.3.1. Pattern

The pattern is a binary representation of the JCC on different days. This feature vector consists of binary data, which are either 0 or 1, where 1 implies the candlestick pattern occurrence on a corresponding day and 0 indicates its non-occurrence. In this research, 14 JCCs were employed and then implemented by using several functions to straightforward the implementation process and recognize the daily candlestick characteristics (Tables 1-3). In these functions, the price is denoted as P. Similarly, the variables CSL (Candle Significance Level) and GSL (Gap Significance Level) are explained as follows:

2.3.2.Vanilla

In this case, the present research only used normalized OHLC prices as the state.

2.3.3. Candle-rep

This state is the daily candlestick representation in the form of a vector with four members, respectively implying the percentage of the upper shadow, the percentage of the lower shadow, the percentage of the main candlestick body, and the candlestick movement direction (i.e., ascending or descending)..

2.3.4. Windowed

Similar to Vanilla, the windowed state represents the normalized OHLC price, with the only difference of using the prices of several consecutive days as the state. The reason for such representation is that JCC pattern recognition usually requires the data of the previous days. Hence, this input state is considered in this section, while using the OHLC price values for three consecutive days to create this state.

2.4. Different components of the DRL-based IA

This research was considered a DRL-based IA, where the DNN acts to predict the right action for the IA based on the input state. Then, the IA performs transactions according to the action predicted by the DNN and the policies learned by the Q-Learning algorithm. Based on the DRL's conventional methods, Replay-Memory and Mini-Batch were used to train the DRL-IA In DQN training methods, a DNN is generally specified separately from the online DNN on the target line. After training a state, the online DNN stores four state values, the reward amount, the action performed, and the next state in the Replay-Memory. Figure 1 depicts the general structure of this DRL-based IA.

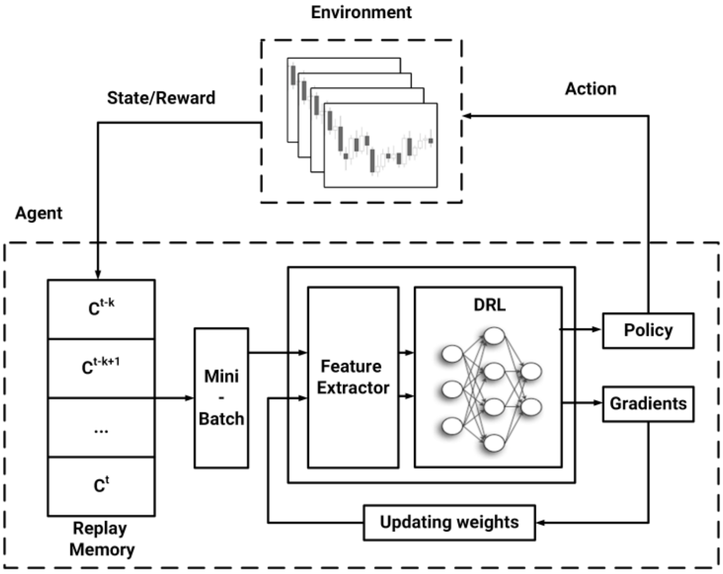

Figure 1. The structure of the DRL-based IA

This model has no feature extractor, and the input state is directly linked to the decision maker's neural network. These structures are discussed in (20) by Taghian et al., with Figure 2 depicting the simple approach in the feature extraction section.

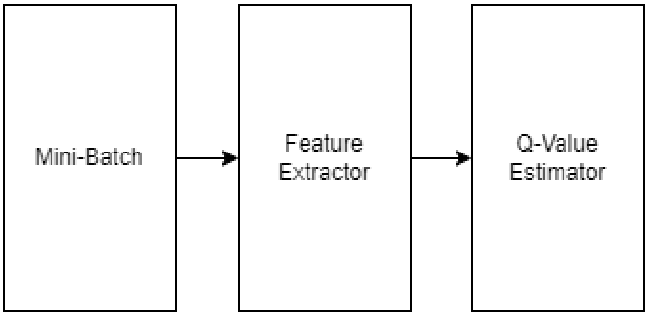

Figure 2. The process of feature extraction in the work by Taghian et al. (20)

The innovative model in the present research alters this structure to check variations in their profitability. Based on the obtained results, the present research adds some structures to the above structures to assess profitability. Figure 3 depicts the general structure of these proposed methods.

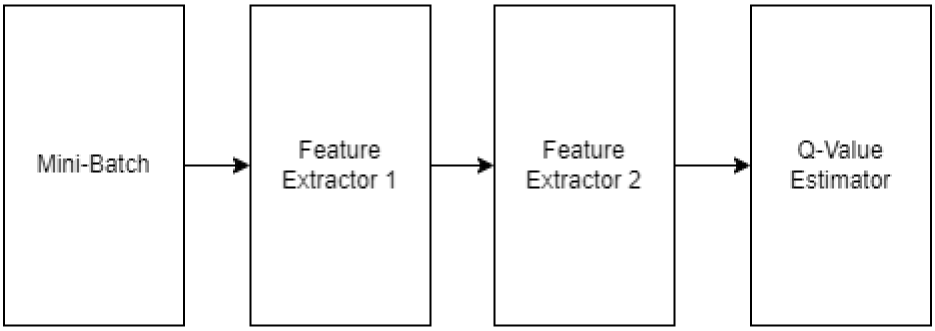

Figure 3. The proposed feature extraction structure

According to the structure given in Figure 3, feature extraction was performed as follows:

2.4.1. CNN1D-MLP

In this structure, the state before entering the decision-making section is first processed by a CNN1D layer and then by the proposed MLP structure.

2.4.2. MLP-CNN1D

Here, the state passes through the MLP network and is then processed by CNN1D. The output serves as the input for the decision-making section.

2.4.3. CNN2D-MLP

This structure is similar to structure 1, with the only difference of CNN2D that substitutes CNN1D.

2.4.4. MLP-CNN2D

This structure is similar to structure 1, with the only difference of CNN2D that substitutes CNN1D.

2.4.5. GRU-MLP

Here, the state first passes through the GRU recurrent neural network (RNN) and then enters the MLP neural network. The output serves as an input for the decision-making network.

2.4.6. MLP-GRU

In this structure, the state is first processed by MLP and then by the GRU network. The output is then used as an input for the decision-making network.

2.4.7. LSTM

Here, an LSTM layer is employed to process the input state. Contrary to other states, this structure has only one feature extractor, similar to those proposed by Taghian et al.

2.4.8. LSTM-MLP

This structure is similar to structure 6, with the only difference of LSTM that substitutes GRU.

2.4.9. MLP-LSTM

This structure is similar to structure 7, with the only difference of LSTM that substitutes GRU.

2.5. The training environment and testing models

The ML models were trained and tested using the Scikit-learn (sklearn) library version 1.2.2 by creating a Python virtual environment on the Windows operating system and the 11th Gen Intel® Core(TM) i5-11260H@2.60Hz processor.

Likewise, the DRL models were trained and tested using PyTorch library version 1.9. in the WSL (windows subsystem for Linux) virtual system and in the Anaconda3 virtual environment. These functions were carried out on the NVIDIA GeForce RTX 3050 graphics processor with the help of the Cuda library, version 11.

3. Findings

3.1. Results of ML models

3.1.1. The aud/usd currency pair

Table 1 presents the most influential features and their scores for the aud/usd currency pair. For this currency pair, the indices Relative Strength Index (RSI), Williams %R oscillator, and stochastic oscillator have played a more critical role than other indices in detecting the trend based on the past information of this currency pair.

Table 1. Features selected for the currency pair aud/usd

Feature	score
RSI(STEP_2)_9	0.062
RSI(STEP_1)_5	0.052
%R_12	0.05
%D_14_3	0.046
%R_14	0.045
RSI(STEP_1)_9	0.043
%R_9	0.042
%K_14	0.029
-DI_10	0.028
%K_21	0.028
+DI_10	0.027
%R_26	0.026
RSI(STEP_1)_12	0.026
%D_21_5	0.025
RSI(STEP_2)_5	0.024
RSI(STEP_2)_12	0.022
RSI(STEP_1)_14	0.019
%R_5	0.017
DX_10	0.016
RSI(STEP_2)_14	0.014
+DI_14	0.013
RSI(STEP_2)_26	0.011
CCI_10	0.01
-DI_14	0.01
+DI_20	0.01
-DI_20	0.01
ADX_10	0.008
MACD_12_26	0.008
DX_14	0.008
-DI_30	0.007
+DI_30	0.007
CCI_14	0.007
DX_20	0.007
kumo_9_12_26	0.006
ADX_14	0.006
CCI_20	0.006
kumo_9_26_52	0.006
ATR_5	0.006
Body	0.006
DX_30	0.006

CCI_30	0.006
Volume	0.006
RSI(STEP_1)_26	0.006
STD_20	0.006
ADX_30	0.006
ATR_9	0.006
ADX_20	0.005
-DM	0.005
+DM	0.005
ATR_14	0.005
ATR_26	0.005
ATR_12	0.005
Chikou_span_26	0.004
Upper_Band_20_2	0.004
EMA_5	0.004
SAR_0.02_0.2	0.004
senkou_span_b_52	0.004
Chikou_span_12	0.004
Highest_High_21	0.004
Open	0.004
Lower_Band_20_2	0.004

Table 2 presents the best scores chosen for ML parameters for the currency pair aud/usd. These parameters are used to develop the best model regarding the accuracy of calculating the profitability of transactions made based on the used strategy.

Table 2. The best settings of models for the currency pair aud/usd

Model	Setting
Decision tree	max_depth=None
Gradient Boosting	n_estimators=200
Gaussian Naïve Bayes	-
K-nearest neighbor	n_neighbors=7
Logistic Regression	C=10, penalty=l2, solver=lbfgs, max_iter=1000
Multilayer Perceptron	hidden_layer_sizes=(128, 128, 128), activation=tanh, solver=adam, batch_size=128, learning_rate=adaptive, max_iter=1000, shuffle=False, early_stopping: True
Random forest	n_estimators=200
SVM	C=50, kernel=rbf

Based on the values of parameters in Table 2, the most accurate ML models are first created and then assessed based on the test data for their profitability in the currency pair aud/usd. As given in Table 3, the majority of these models are highly accurate and satisfactorily profitable within the test period. However, the decision tree is the most profitable model among all ML models.

Table 4. The results of ML models for the currency pair aud/usd

Model	Accuracy	RoA rate	RoA volume	Mean daily profit
Decision tree	91.74	**25.55**	**255.56**	**0.36**
Gradient Boosting	**93.88**	21.27	212.73	0.3
Gaussian Naïve Bayes	68.84	13.38	123.81	0.19
K-nearest neighbor	74.53	11.5	115.07	0.16
Logistic Regression	75.96	10.32	103.28	0.14
Multilayer Perceptron	89.61	9.88	98.81	0.14
Random forest	88.9	22.05	220.59	0.31
SVM	84.63	6.16	61.69	0.08

3.1.2. The eur/usd currency pair

For this currency pair, the indices RSI, Williams %R oscillator, and ADX have played a more critical role than other indices in detecting the trend. It can be seen that RSI has been the most influential for trend detection in steps 1 and 2 for nine days.

Table 4 presents the best values of ML parameters for the currency pair eur/usd. These parameters are used to develop the best model regarding the accuracy of calculating the profitability of transactions made based on the used strategy.

Table 4. The best settings of models for the currency pair eur/usd

Model	Setting
Decision tree	max_depth=20
Gradient Boosting	n_estimators=200
Gaussian Naïve Bayes	-
K-nearest neighbor	n_neighbors=7
Logistic Regression	C=100, penalty=l2, solver=saga, max_iter=1000
Multilayer Perceptron	hidden_layer_sizes=(64, 64, 64), activation=relu, solver=adam, batch_size=64, learning_rate=adaptive, max_iter=1000, shuffle=False, early_stopping: True
Random forest	n_estimators=10
SVM	C=100, kernel=rbf

Based on the values of parameters in Table 4, the most accurate ML models are first created and then assessed based on the test data for their profitability in the currency pair eur/usd. As given in Table 5, the majority of these models are highly accurate and satisfactorily profitable within the test period. However, the Multilayer Perceptron is the most profitable model among all ML models.

Table 5. The results of ML models for the currency pair eur/usd

Model	Accuracy	RoA rate	RoA volume	Mean daily profit
Decision tree	89.18	6.14	61.4	0.08
Gradient Boosting	**93.59**	7.29	72.94	0.1

Gaussian Naïve Bayes	70.27	-0.45	-4.5	0
K-nearest neighbor	74.25	7.62	76.25	0.1
Logistic Regression	76.81	4.58	45.88	0.06
Multilayer Perceptron	88.47	**13.2**	**132.02**	**0.18**
Random forest	86.34	7.52	75.2	0.01
SVM	85.63	4.9	49	0.06

3.1.3. The gbp/usd currency pair

For this currency pair, the indices RSI, Stochastic oscillator, Williams %R oscillator, ADX, and DI have played a more critical role than other indices in detecting the trend.

Table 6 presents the best values of ML parameters for the currency pair gbp/usd. These parameters are used to develop the best model regarding the accuracy of calculating the profitability of transactions made based on the used strategy.

Table 6. The best settings of models for the currency pair gbp/usd

Model	Setting
Decision tree	max_depth=100
Gradient Boosting	n_estimators=200
Gaussian Naïve Bayes	-
K-nearest neighbor	n_neighbors=7
Logistic Regression	C=100, penalty=l2, solver=lbfgs, max_iter=1000
Multilayer Perceptron	hidden_layer_sizes=(128), activation=relu, solver=adam, batch_size=64, learning_rate=adaptive, max_iter=1000, shuffle=False, early_stopping: True
Random forest	n_estimators=50
SVM	C=25, kernel=rbf

Based on the values of parameters in Table 6, the most accurate ML models are first created and then assessed based on the test data for their profitability in the currency pair gbp/usd. As given in Table 7, the majority of these models are highly accurate and satisfactorily profitable within the test period. However, the Logistic Regression (with a 20% profit) is the most profitable model among all ML models.

Table 7. The results of ML models for the currency pair gpb/usd

Model	Accuracy	RoA rate	RoA volume	Mean daily profit
Decision tree	89.47	7.87	78.7	0.11
Gradient Boosting	**92.6**	4.76	47.69	0.06
Gaussian Naïve Bayes	69.98	8.08	80.89	0.11
K-nearest neighbor	71.26	15.91	159.15	0.22
Logistic Regression	74.96	**21.11**	**201.19**	**0.28**
Multilayer Perceptron	87.33	14.17	140.79	0.2
Random forest	85.91	18.19	181.97	0.25
SVM	82.07	16.69	166.96	0.23

3.1.4. The usd/cad currency pair

For this currency pair, the indices RSI, Stochastic oscillator, Williams %R oscillator, and ADX have played a more critical role than other indices in detecting the trend.

Table 8 presents the best values of ML parameters for the currency pair usd/cad. These parameters are used to develop the best model regarding the accuracy of calculating the profitability of transactions made based on the used strategy.

Table 8. The best settings of models for the currency pair usd/cad

Model	Setting
Decision tree	max_depth=50
Gradient Boosting	n_estimators=100
Gaussian Naïve Bayes	-
K-nearest neighbor	n_neighbors=30
Logistic Regression	C=0.7, penalty=l2, solver=lbfgs, max_iter=1000
Multilayer Perceptron	hidden_layer_sizes=(128, 128), activation=tanh, solver=adam, batch_size=256, learning_rate=adaptive, max_iter=1000, shuffle=False, early_stopping: True
Random forest	n_estimators=50
SVM	C=100, kernel=rbf

Based on the values of parameters in Table 8, the most accurate ML models are first created and then assessed based on the test data for their profitability in the currency pair usd/cad. As given in Table 9, despite a high accuracy, these models have often resulted in losses, making trend detection a fairly daunting task. However, the Gradient Boosting model (with a 10% profit) is the most profitable model among all ML models.

Table 9. The results of ML models for the currency pair usd/cad

Model	Accuracy	RoA rate	RoA volume	Mean daily profit
Decision tree	88.05	-0.83	-8.37	-0.01
Gradient Boosting	92.03	7.04	70.45	0.1
Gaussian Naïve Bayes	72.68	-2.03	-20.34	-0.02
K-nearest neighbor	73.82	-1.18	-11.8	-0.01
Logistic Regression	77.66	0.22	2.29	0
Multilayer Perceptron	89.47	3.61	36.13	0.05
Random forest	87.48	0.27	2.71	0
SVM	86.06	-1.28	-12.82	-0.01

3.2. The results for DRL-based IAs

3.2.1. The aud/usd currency pair

According to Table 10, the structures based on CNN and GRU algorithms have been profitable, while DQN-based models (except for the input state of Vanilla with a gamma value of 0.8) have caused losses and failed to perform satisfactorily. The MLP model (with inputs Vanilla and Windowed) has been profitable and demonstrated satisfactory performance. However, this model has caused losses for two input states of Pattern and Candle-Rep.

Table 10. The results of the previous DRL-based IAs for the currency pair aud/usd

IA	Gamma value	RoA rate	RoA volume	Mean daily profit
DQN-Pattern	0.9	-4.85	-48.57	-0.07
DQN-Pattern	0.8	-3.19	-31.92	-0.04
DQN-Vanilla	0.9	-1.3	-13.01	-0.01
DQN-Vanilla	0.8	43.18	431.18	0.63
DQN-Candle-Rep	0.9	-4.83	-48.33	-0.07
DQN-Candle-Rep	0.8	-4.83	-48.33	-0.07
DQN-Windowed	0.9	-4.19	-41.93	-0.06
DQN-Windowed	0.8	21.09	210.9	0.31
MLP-Pattern	0.9	-5.02	-50.22	-0.07
MLP-Pattern	0.8	0	0	0
MLP-Vanilla	0.9	53.6	536.25	0.79
MLP-Vanilla	0.8	52.5	525.07	0.77
MLP-Candle-Rep	0.9	-5.02	-50.22	-0.07
MLP-Candle-Rep	0.8	-4.8	-48.33	-0.07
MLP-Windowed	0.9	46.04	460.49	0.68
MLP-Windowed	0.8	37.6	376.6	0.55
CNN1D-Windowed	0.9	18.4	184.86	0.27
CNN1D-Windowed	**0.8**	**56.1**	**561.27**	**0.82**
CNN2D-Windowed	0.9	40.7	407	0.6
CNN2D-Windowed	0.8	15.5	155.82	0.23
GRU-Windowed	0.9	0.5	5.28	0.007
GRU-Windowed	0.8	24.9	249.92	0.36

Based on Table 11, the CNN- and MLP-based structures deliver marked profit. Similarly, the simple LSTM structure (with a gamma value of 0.8) has been profitable. Merging the GRU and MLP models has not been profitable and, instead, caused some losses. Likewise, merging LSTM with MLP has not been significantly profitable.

Table 11. The results of the proposed DRL-based IA for the currency pair aud/usd

IA	Gamma value	RoA rate	RoA volume	Mean daily profit
CNN1D-MLP	0.9	28.6	284.24	0.4
CNN1D-MLP	**0.8**	**46.9**	**649.98**	**0.92**
MLP-CNN1D	0.9	28.6	286.24	0.4
MLP-CNN1D	0.8	64.9	649.98	0.92
CNN2D-MLP	0.9	49.3	493.74	0.7
CNN2D-MLP	0.8	46.2	462.04	0.65
MLP-CNN2D	0.9	49.3	493.74	0.7
MLP-CNN2D	0.8	46.2	462.04	0.65
GRU-MLP	0.9	-5.28	-52.84	-0.07
GRU-MLP	0.8	7.88	78.84	0.11
MLP-GRU	0.9	-5.28	-52.84	-0.07

MLP-GRU	0.8	7.88	78.84	0.11
LSTM	0.9	-5.28	-52.84	-0.07
LSTM	0.8	43.11	431.16	0.61
LSTM-MLP	0.9	-5.28	-52.84	-0.07
LSTM-MLP	0.8	7.88	78.84	0.11
MLP-LSTM	0.9	-5.28	-52.84	-0.07
MLP-LSTM	0.8	7.88	78.84	0.11

3.2.2. The eur/usd currency pair

According to Table 12, the structures based on CNN and GRU algorithms have been profitable, except for one input state in the GRU model with a gamma value of 0.9 that has caused losses. The DQN models (except for the inputs Vanilla and Windowed) have not been profitable (with a gamma value of 0.8) and demonstrated unsatisfactory performance. The MLP model (with inputs Vanilla and Windowed) has been profitable and demonstrated satisfactory performance. However, this model has caused losses for two input states of Pattern and Candle-Rep.

Table 12. The results of the previous DRL-based IAs for the currency pair eur/usd

IA	Gamma value	RoA rate	RoA volume	Mean daily profit
DQN-Pattern	0.9	-8.6	-86.48	-12.77
DQN-Pattern	0.8	-9.4	-94.29	-0.13
DQN-Vanilla	0.9	1.6	1.68	0.002
DQN-Vanilla	0.8	25.2	252.41	0.37
DQN-Candle-Rep	0.9	-11	-110.01	-0.16
DQN-Candle-Rep	0.8	-8.	-86.48	-0.12
DQN-Windowed	0.9	-2.3	-23.24	-0.03
DQN-Windowed	0.8	13.09	130.98	0.19
MLP-Pattern	0.9	0	0	0
MLP-Pattern	0.8	-8.6	-86.77	-0.12
MLP-Vanilla	0.9	23.2	232.49	0.34
MLP-Vanilla	0.8	21.1	211.36	0.31
MLP-Candle-Rep	0.9	0	0	0
MLP-Candle-Rep	0.8	-.94	-94.91	-0.14
MLP-Windowed	0.9	20.01	200.17	0.29
MLP-Windowed	0.8	24.9	249.11	0.36
CNN1D-Windowed	0.9	0.005	0.05	0
CNN1D-Windowed	**0.8**	**35.8**	**358.72**	**0.52**
CNN2D-Windowed	0.9	18.4	148.88	0.27
CNN2D-Windowed	0.8	28.6	286.2	0.42
GRU-Windowed	0.9	-4.1	-41.43	-0.06
GRU-Windowed	0.8	14.5	145.32	0.21

Based on Table 13, the CNN- and MLP-based structures are profitable, with the profit that is higher for the 2D processing than the 1D processing. Similarly, the simple LSTM structure (with a gamma value of 0.8) has been more profitable than other structures and resulted in the highest profit. Merging the GRU and MLP models has not been profitable and, instead, caused some losses, even for the gamma value of 0.8.

Table 13. The results of the proposed DRL-based IA for the currency pair eur/usd

IA	Gamma value	RoA rate	RoA volume	Mean daily profit
CNN1D-MLP	0.9	0.89	8.91	0.01
CNN1D-MLP	0.8	13.59	135.95	0.19
MLP-CNN1D	0.9	0.89	8.91	0.01
MLP-CNN1D	0.8	13.59	135.95	0.19
CNN2D-MLP	0.9	18.8	188.07	0.26
CNN2D-MLP	0.8	24.78	247.89	0.35
MLP-CNN2D	0.9	18.8	188.07	026
MLP-CNN2D	0.8	24.78	247.89	0.35
GRU-MLP	0.9	8.94	89.4	0.12
GRU-MLP	0.8	-9.21	-92.16	-0.13
MLP-GRU	0.9	8.94	89.4	0.12
MLP-GRU	0.8	-9.21	-92.16	-0.013
LSTM	**0.9**	**38.55**	**385.54**	**0.54**
LSTM	0.8	16.16	161.64	0.22
LSTM-MLP	0.9	8.94	89.4	0.12
LSTM-MLP	0.8	-9.21	-92.16	-0.13
MLP-LSTM	0.9	8.94	89.4	0.12
MLP-LSTM	0.8	-9.21	-92.16	-0.13

3.2.3. The gbp/usd currency pair

According to Table 14, the structures based on CNN and GRU algorithms have been profitable. Similarly, the DQN models (with the inputs Vanilla and Windowed) have been profitable and demonstrated satisfactory performance. The MLP model (with inputs Vanilla and Windowed) has been profitable and performed satisfactorily. However, this model has caused losses for two input states of Pattern and Candle-Rep.

Table 14. The results of the previous DRL-based IAs for the currency pair gbp/usd

IA	Gamma value	RoA rate	RoA volume	Mean daily profit
DQN-Pattern	0.9	-6.4	-64.52	-0.09
DQN-Pattern	0.8	4.4	44.17	0.06
DQN-Vanilla	0.9	29.7	297.83	0.43
DQN-Vanilla	0.8	34.1	341.03	0.5
DQN-Candle-Rep	0.9	-6.4	-64.52	-0.09
DQN-Candle-Rep	0.8	-10.4	-104.03	-0.15
DQN-Windowed	0.9	34.6	346.04	0.51
DQN-Windowed	0.8	19.7	197.15	0.29
MLP-Pattern	0.9	-6.4	-64.52	-0.09
MLP-Pattern	0.8	-3.4	-34.83	-0.09
MLP-Vanilla	**0.9**	**39.4**	**394.87**	**0.58**
MLP-Vanilla	0.8	39.3	393.44	0.58
MLP-Candle-Rep	0.9	-6.4	-64.61	-0.09
MLP-Candle-Rep	0.8	-6.4	-64.52	-0.09
MLP-Windowed	0.9	39.1	391.14	0.57

MLP-Windowed	0.8	35.9	359.8	0.531
CNN1D-Windowed	0.9	7.7	77.48	0.11
CNN1D-Windowed	0.8	37.5	375.97	0.55
CNN2D-Windowed	0.9	25.5	255.35	0.37
CNN2D-Windowed	0.8	27.4	274.36	0.4
GRU-Windowed	0.9	28.8	288.34	0.42
GRU-Windowed	0.8	25.3	253.31	0.37

Based on Table 15, the CNN- and MLP-based structures are profitable, with the profit that is higher for the 1D processing than the 2D processing. Similarly, the simple LSTM structure (with a gamma value of 0.9) has been markedly profitable and performed satisfactorily. Merging the GRU and LSTM models with the MLP model has not been significantly profitable and, instead, caused some losses, even for the gamma value of 0.9.

Table 15. The results of the proposed DRL-based IA for the currency pair gbp/usd

IA	Gamma value	RoA rate	RoA volume	Mean daily profit
CNN1D-MLP	**0.9**	**37.91**	**379.18**	**0.53**
CNN1D-MLP	0.8	36.22	363.21	0.51
MLP-CNN1D	0.9	37.91	379.18	0.53
MLP-CNN1D	0.8	36.32	363.21	0.51
CNN2D-MLP	0.9	29.99	299.96	0.42
CNN2D-MLP	0.8	35	350.04	049
MLP-CNN2D	0.9	29.99	299.96	0.42
MLP-CNN2D	0.8	35	350.04	0.49
GRU-MLP	0.9	-6.26	-62.66	0.08
GRU-MLP	0.8	24.46	244.67	0.34
MLP-GRU	0.9	-6.26	-62.66	-0.08
MLP-GRU	0.8	24.46	244.67	0.34
LSTM	0.9	35.94	359.46	0.51
LSTM	0.8	25.92	259.26	0.36
LSTM-MLP	0.9	0.9	-6.26	-62.66
LSTM-MLP	0.8	0.8	24.46	244.67
MLP-LSTM	0.9	0.9	-6.26	-62.66
MLP-LSTM	0.8	0.8	24.46	244.67

3.2.4. The usd/cad currency pair

According to Table 16, the structures based on CNN and GRU algorithms have been profitable. Contrarily, the DQN models (except for the state Windowed) have shifted toward not transaction or delivered trivial profit. The MLP model (with inputs Vanilla and Windowed) has been profitable and performed satisfactorily. However, this model has caused losses and/or biased for two input states of Pattern and Candle-Rep.

When assessing the input states (similar to previous currency pairs), it is evident that the states Vanilla and Windowed are potentially profitable based on the DNN structure. However, the states Pattern and Candle-Rep fail to properly convey the environment features to the IA and thus perform unsatisfactorily.

Table 16. The results of the previous DRL-based IAs for the currency pair usd/cad

IA	Gamma value	RoA rate	RoA volume	Mean daily profit
DQN-Pattern	0.9	5.4	54.13	0.07
DQN-Pattern	0.8	8.3	83.85	0.12
DQN-Vanilla	0.9	0	0	0
DQN-Vanilla	0.8	4.6	46.88	0.06
DQN-Candle-Rep	0.9	0	0	0
DQN-Candle-Rep	0.8	0	0	0
DQN-Windowed	0.9	20.2	202.81	0.29
DQN-Windowed	0.8	21.4	214.63	0.31
MLP-Pattern	0.9	0	0	0
MLP-Pattern	0.8	0	0	0
MLP-Vanilla	0.9	34.9	349.65	0.51
MLP-Vanilla	0.8	26.3	263.07	0.38
MLP-Candle-Rep	0.9	-0.1	-1.53	0.002
MLP-Candle-Rep	0.8	-0.01	-0.13	0
MLP-Windowed	0.9	2.7	270.92	0.4
MLP-Windowed	0.8	29.7	297.91	0.44
CNN1D-Windowed	0.9	21.3	213.86	0.31
CNN1D-Windowed	**0.8**	**48**	**480.02**	**0.7**
CNN2D-Windowed	0.9	21.1	210.17	0.03
CNN2D-Windowed	0.8	20.6	206.52	0.3
GRU-Windowed	0.9	20.7	207.23	0.3
GRU-Windowed	0.8	28.8	288.5	0.42

Based on Table 17, the CNN- and MLP-based structures are profitable. However, for the gamma value of 0.9, the CNN2D-MLP model has been biased and failed to achieve the optimum policy and profit. This profit is higher for the 1D processing than the 2D processing. The simple LSTM structure has not been profitable alone, but its combination with MLP has resulted in marked profit compared to other models. Merging the GRU and LSTM models with the MLP model has been significantly profitable for this currency pair. Thus, these structures can be used in transactions.

Table 17. The results of the proposed DRL-based IA for the currency pair usd/cad

IA	Gamma value	RoA rate	RoA volume	Mean daily profit
CNN1D-MLP	**0.9**	**38.96**	**389.68**	**0.55**
CNN1D-MLP	0.8	30.92	309.2	0.43
MLP-CNN1D	0.9	38.96	389.68	0.55
MLP-CNN1D	0.8	30.92	309.2	0.43
CNN2D-MLP	0.9	0	0	0
CNN2D-MLP	0.8	31.05	310.56	0.44
MLP-CNN2D	0.9	0	0	0
MLP-CNN2D	0.8	31.05	310.56	0.44
GRU-MLP	0.9	37.11	371.12	0.52
GRU-MLP	0.8	32.65	326.56	0.46
MLP-GRU	0.9	37.11	371.12	0.52
MLP-GRU	0.8	32.65	326.56	0.46
LSTM	0.9	3.15	31.58	0.04
LSTM	0.8	0	0	0

LSTM-MLP	0.9	31.11	371.12	0.52
LSTM-MLP	0.8	32.65	326.56	0.46
MLP-LSTM	0.9	37.11	371.12	0.52
MLP-LSTM	0.8	32.65	326.56	0.46

Performance comparison of the best IAs between previous structures and the proposed structure

As shown in Figure 4 for the currency pair aud/usd, the gamma value of 0.8 outperforms the gamma value of 0.9 in profitability. Correspondingly, the transactions with the gamma value of 0.8 have properly enhanced the value of the trader's stock portfolio. The proposed structure has markedly outperformed the best previous structures in delivering profit.

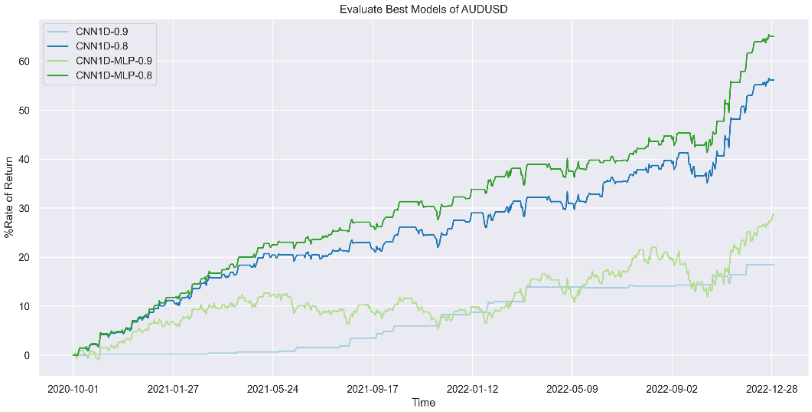

Figure 4. Transactions of the best IAs in the currency pair aud/usd

As shown in Figure 5 for the currency pair eur/usd, the gamma value of 0.8 (for the structure CNN1D) and the gamma value of 0.9 (for the structure LSTM) have been profitable. Correspondingly, the transactions with these gamma values have properly enhanced the value of the trader's stock portfolio. The proposed structure has (for some percentages) outperformed the best previous structures in delivering profit.

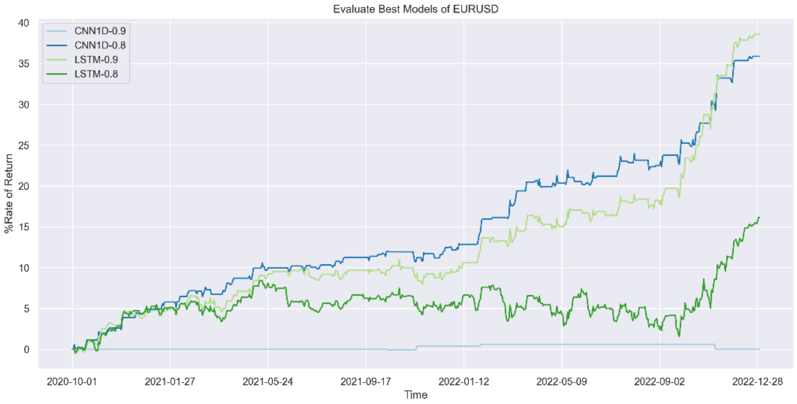

Figure 5. Transactions of the best IAs in the currency pair eur/usd

As shown in Figure 6 for the currency pair gbp/usd, the MLP-Vanilla structure has resulted in nearly equal profit for both gamma values of 0.8 and 0.9. Although the best proposed model (i.e., CNN1D-MLP) has resulted in satisfactory profit for both gamma values, the profit from this structure is less than the profit obtained from the best previous structures.

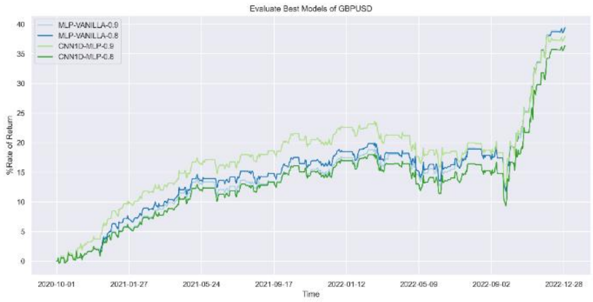

Figure 6. Transactions of the best IAs in the currency pair gbp/usd

As shown in Figure 7 for the currency pair usd/cad, the gamma value of 0.8 has delivered higher profit than the gamma value of 0.9 for the structure CNN1D. Our proposed structure (CNN1D-MLP) has resulted in a higher profit (by 10%) in the gamma value of 0.8 than the gamma value of 0.9. Although both structures are properly profitable, CNN1D (with a gamma value of 0.8) has resulted in the greatest profit, compared to the best previous structures.

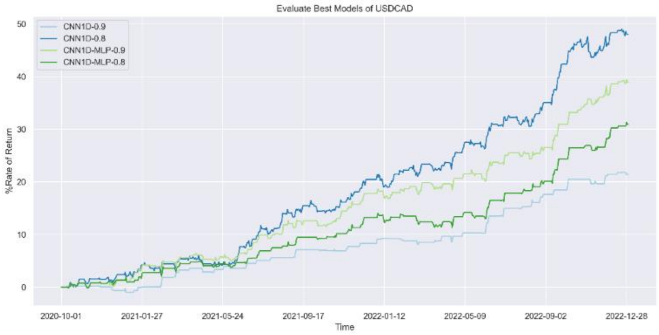

Figure 7. Transactions of the best IAs in the currency pair usd/cad

Table 18. Comparing the best ML and DRL models

Currency pair	ML model	Final profit of the ML model	DRL model	Final profit of the DRL model
aud/usd	Decision tree	255.56	CNN1D-MLP(γ=0.8)	649.98
eur/usd	Multilayer perceptron	132.04	LSTM(γ=0.9)	385.54
gbp/usd	Logistic regression	201.19	MLP-Vanilla(γ=0.9)	394.87
usd/cad	Gradient boosting	70.45	CNN1D-Windowed(γ=0.8)	480.02

As with Table 18, DRL models have outperformed (and are more profitable than) ML models for all currency pairs. Furthermore, the models proposed in this research have outperformed the previously implemented models in two currencies. Although other structures fail to achieve the highest profit, they have resulted in marked profit compared to the previous structures. Thereby, DNN structures are recommended to be used in the daily timeframe for making transactions. Additionally, by comparing the results of the proposed structures, it is worth mentioning that all strategies of merging models have equally been effective in enhancing profitability. As such, CNN1D-MLP and MLP-CNN1D structures have resulted in equal profit in the proposed model.

Conclusion

This research proposed a profitable DNN-based IA that can trade in the Forex market and deliver a marked profit within a certain period. In this research, DRL models were generally much more profitable than the ML models and the proposed strategy. Such profitability is constant for diverse currency pairs for some structures and states, implying that these structures and states are generally suitable for training and profiting in other periods. Indeed, the chance of their profitability is high for other currencies and periods. By contrast, some structures and states have caused losses in nearly most cases instead of making profits and can be overlooked in further research. Accordingly, future research is advised to merely concentrate on profitable structures and states.

Furthermore, future research is recommended to use the outputs of profitable IAs by a voting system and in a combined manner. Indeed, instead of relying on the decision made by a reliable and profitable DRL-IA, the model's profitability is recommended to be enhanced by weighted voting between different DRL-IAs. For this, the decision made by the IA with higher profitability is suggested to receive more weight. The next suggestion is to define diverse states for training on these structures. For instance, the MACD indicator, the corresponding signal line, and the Histogram line can be used as the operating state to introduce the market state. Likewise, the profitability of the structures can be investigated compared to the current state.

Bibliography

[1] D. W. S. Z. Y. F. S. L. Q. Z. Y. Wang, "Deep q-trading," *Cslt.Riit.Tsinghua.Edu.Cn*, pp. 1-9, 2017

[2] X.-Y. L. S. Z. H. Y. Z. Xiong, "A. Walid, Practical deep reinforcement learning approach for stock trading," *arXiv preprint arXiv:1811.07522.*

[3] P. N. V. C. Y. Li, "Application of deep reinforcement learning in stock trading strategies and stock forecasting," *Computing*, pp. 1-18, 2019.

[4] H. G. A. S. D. Van Hasselt, "Deep reinforcement learning with double q-learning," *Phoenix*, vol. 2, p. 5, 2016.

[5] X. L. Z. Z. S. Luo, "A novel cnn-ddpg based ai-trader: Performance and roles in business operations," *Transportation Research Part E: Logistics and Transportation Review*, vol. 131, pp. 68-79, 2019.

[6] N. P. A. T. A. T. K. S. Zarkias, "Deep reinforcement learning for financial trading using price trailing," *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP), IEEE*, pp. 3067-3071, 2019.

[7] S. Z. S. R. Z. Zhang, "Deep reinforcement learning for trading," *The Journal of Financial Data Science*, vol. 2, pp. 25-40, 2020.

[8] D. E. T. Théate, "An application of deep reinforcement learning to algorithmic trading," *arXiv preprint arXiv:2004.06627*, 2020.

[9] H. C. J. W. L. T. V. L. H. F. X. Wu, "Adaptive stock trading strategies with deep reinforcement learning methods," *Information Sciences*, 2020.

[10] T. K. S. M. I. K. P. C. A. Suchaimanacharoen, "Empowered pg in forex trading," *2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), IEEE*, pp. 316-319, 2020.

[11] B. Z. Y. L. M. Y. Y. S. K. Lei, "Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading," *Expert Systems with Applications*, vol. 140, 2020.

[12] Northcott, The complete guide to using candlestick charting: How to earn high rates of return-safely, Atlantic Publishing Company, 2009.

[13] Y. H. K. Y. Z. Y. F. S. Z. Z. F. X. J. L. N. R. T. H. e. a. G. Hu, "Deep stock representation learning: From candlestick charts to investment decisions," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE*, pp. 2706-2710, 2018.

[14] U. M. D. C. a. J. S. Jonathan Masci, "Stacked convolutional auto-encoders for hierarchical feature extraction," *Artificial Neural Networks and Machine Learning–ICANN 2011*, pp. 52-59, 2011.

[15] O. S. S. Thammakesorn, "Generating trading strategies based on candlestick chart pattern characteristics," *Journal of Physics: Conference Series*, vol. 1195, 2019.

[16] G. V. Kass, "An Exploratory Technique for Investigating Large Quantities of Categorical Data," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 29, no. 2, pp. 119-127, 1980.

[17] Orquín-Serrano, "Predictive power of adaptive candlestick patterns in forex market. eurusd case," *Mathematics*, vol. 8, no. 5, p. 802, 2020.

[18] G. T. U. K. S. Birogul, "Yolo object recognition algorithm and "buy-sell decision" model over 2d candlestick charts," *IEEE Access*, vol. 8, pp. 91894-91915, 2020

[19] L. C. D. Fengqian, "An adaptive financial trading system using deep reinforcement learning with candlestick decomposing features," *IEEE Access*, vol. 8, p. 63666–63678, 2020.

[20] A. R. S. Mehran Taghian, "Learning Financial Asset-Specific Trading Rules via Deep Reinforcement Learning," *arXiv preprint arXiv:2010.14194*, 2020.