J. Electrical Systems 20-7s (2024): 3280-3291

Yue Zhang[1*,]
Yuyin Pei [2]

Optimization and Application Exploration of Quantitative Trading Strategy of Reinforcement Learning in Cryptocurrency Trading

Journal of Electrical Systems

***Abstract:* -** The popularity of cryptocurrency markets has significantly increased, which has encouraged many financial traders to pursue huge gains in cryptocurrency trading. Technical analysis and machine learning have also been integrated by some researchers and investors to predict future market patterns. Nevertheless, creating profitable trading strategies is still seen to be a very difficult undertaking, even with the application of these techniques. This manuscript proposes an exploration of quantitative trading strategy of reinforcement learning in cryptocurrency trading with multi-scale fusion self attention generative adversarial network (EQTS-CT-MFSGAN). Initially, the data is collected from Open-High Low-Close-Volume (OHLCV) market data. Afterward, the data's are fed to pre-processing. In pre-processing segment, Affine-Mapping Based Variational Ensemble Kalman Filter (AM-VEnKF) is used to clean the data. The outcome from the pre-processing data is transferred to the MFSGAN. The MFSGAN method is used to classify the cryptocurrency trading such as high risk, low risk and no risk. The Hippopotamus Optimization Algorithm (HOA) is used to optimize the weight parameter of MFSGAN. The proposed technique is implemented in Python and the efficiency of the proposed EQTS-CT-MFSGAN technique is estimated with the help of several performances like accuracy, precision, recall, sensitivity, specificity, F measure and cumulative profit. Proposed SS-ABSR-MFSGAN method attains 21.34%, 23.54% and 23.76% higher accuracy, 21.34%, 21.39% and 20.28% higher precision, 18.82%, 20.53% and 23.79% higher f1-score are analyzed with existing techniques like Deep reinforcement learning for the optimal placement of cryptocurrency limit orders (OPCLO-DDQN), UNSURE-A machine learning approach to cryptocurrency trading (UCT-TCN) and Multi-Agent Deep Reinforcement Learning With Progressive Negative Reward for Cryptocurrency Trading (PNRCT-MAPPO) respectively.

Keywords: Cryptocurrency Trading, Hippopotamus Optimization Algorithm, Multi-Scale Fusion Self Attention Generative Adversarial Network, Reinforcement Learning, Quantitative Trading.

I. INTRODUCTION

a) Background

Cryptocurrencies, virtual currencies that operate independently across computer networks, are decentralized and not under the control of any single entity [1]. With their exponential rise in popularity and value in recent years, traders and investors have increasingly focused on digital assets like Bitcoin [2]. Cryptocurrency assets are traded similarly to stocks, despite their underlying differences [3]. For instance, Bitcoin markets operate non-stop, 24/7, and with no intermediaries involved in transactions, there's potential for reduced transaction fees [4]. These factors contribute to the allure of cryptocurrency markets, offering traders lucrative opportunities, albeit with increased risk [5]. The abundance of financial market information has led many traders and investors to employ technical analysis, a method utilizing historical market data to predict future movements [6, 7]. Technical indicators, representing an asset's momentum, volatility, and trend patterns, are commonly utilized [8]. However, reliance solely on technical indicators can lead to erroneous trend predictions, prompting financial analysts and investors to employ a combination of indicators. Thanks to blockchain technology, cryptocurrencies can be exchanged directly among users without intermediary interference, rendering them immune to manipulation by governments or organizations. This characteristic contributes to the intrinsic value of cryptocurrencies, despite lacking physical representation.

b) Challenges

Technical analysis is the main instrument used by traders to find winning chances. To predict future market movements, several researchers and investors have also blended machine learning with technical analysis. But even with the application of these techniques, creating profitable trading strategies is still seen to be a very difficult undertaking. Navigating the complexities of cryptocurrency trading, particularly within the realm of quantitative strategies driven by reinforcement learning, presents multifaceted challenges. Primarily, the quality and availability of data pose significant hurdles, given the decentralized and often opaque nature of

[1] [1] Southwest Jiaotong University Hope College, Chengdu, Sichuan, 610400, China
[2] Chengdu Rail Transit Resource Operation and Management Co., Ltd, Chengdu, Sichuan, 610000, China
*Corresponding author e-mail: ZY15680062586@126.com

cryptocurrency markets. Volatility, inherent to these markets, demands robust risk management frameworks to safeguard against sudden price swings. Furthermore, the risk of model overfitting looms large, necessitating rigorous validation procedures to ensure generalizability. Market manipulation, a persistent concern, underscores the need for strategies resilient to such malfeasance.

Regulatory uncertainty adds another layer of complexity, requiring strategies that are adaptable to evolving legal landscapes. Computational complexity, coupled with the imperative for interpretability and explainability, further complicates the development and deployment of effective trading strategies. Addressing these challenges demands a nuanced understanding of both cryptocurrency dynamics and advanced quantitative techniques, alongside a commitment to continuous adaptation and compliance adherence.

c) Literature Review

A few recent studies are covered below. A number of research projects were proposed in the literature about reinforcement learning in bitcoin trading.

Matthias [9] have created the first widely used DRL to optimise the placement of limit orders at bitcoin exchanges. A virtual limit order exchange is used to reward agents based on the realised shortage across a sequence of time steps for both training and out-of-sample assessment. Produce features that tell the agent about the present status of the market based on the literature. Examine experimentally how the latest DRL algorithms perform against many benchmarks utilising more than 3.5 million order book states and 18 months of high-frequency data from significant currency pairs and exchanges. When proximal policy optimisation is used instead of deep double Q-networks and other benchmarks, one may consistently acquire better order placement techniques. Additional examination clarified the mystery behind the learned execution technique.. Queue imbalances and current liquidity costs were important features; the latter can be utilised to forecast short-term mid-price returns.

Kochliaridis et al. [10] have developed the unsure - a machine learning approach to crypto currency trading. Trading crypto currencies can be very profitable, but because of the sharp price swings and high level of market noise, there were also considerable risks involved. Traders generally employ a variety of forecasting techniques, including technical analysis and machine learning, to maximize profits and reduce risks. Making profitable trading methods in noisy markets was still a challenging task, though. DRL agents have recently demonstrated remarkable performance on difficult tasks, such algorithmic trading; nevertheless, to properly train them, a substantial amount of labour and high-quality data are needed.. DRL agents were also less appealing to traders since they were not explainable.

Kumlungmak and Vateekul [11] have developed PNRCT-MAPPO. Reinforcement learning has been used recently to trade cryptocurrency profitably. However, the market's volatility, particularly during gloomy times, made trading cryptocurrencies an extremely difficult undertaking. As a result, the effectiveness of reinforcement learning techniques for trading cryptocurrencies in the body of current work was limited.A MAPPO-based cryptocurrency trading technique that optimises the individual and group performance of the agents using a cooperative multi-agent scheme and a local-global reward function. Agents with a progressive penalty were trained utilising both a multi-objective optimisation approach and a multi-scale continuous loss (MSCL) incentive to prevent repeated losses of portfolio value.

Su [12] have developed a analysis of cryptocurrency and its combination with quantitative transactions. Due to its unique characteristics and inherent value, a significant number of investors frequently mix digital currencies with quantitative trading in order to profit from the concepts and generate additional returns.

Yu et al. [13] have developed an introduction to the value and trading technique of cryptocurrencies: the bitcoin gold, litecoin silver. In the past, conventional monetary systems have assigned different duties to gold and silver. Silver has long been used as a medium of exchange, but gold has historically been seen as a better store of value, which has prompted people to hoard it. A new paradigm of value and exchange has emerged with the growth of cryptocurrencies in the financial world. Still, not much was known about these digital assets' store-of-value quality. Litecoin's inventor, Charlie Lee, previously compared Bitcoin to gold and Litecoin to silver. Our analysis uses a number of indicators, such as Coin Days Destroyed, Unspent Transaction Outputs, Weighted Average Lifespan, Spent Transaction Outputs, and Public On-Chain Transaction Data to validate this parallel.

Fang et al. [14] have developed the cryptocurrency market in an environment with high frequency. Some characteristics were shared by all cryptocurrencies and help them beat asset-specific models. Additionally, demonstrate that feeding machine learning models with lengthy data point sequences was not very useful as predictions remain unchanged. In addition, to overcome the technological difficulty of creating a lean predictor that works effectively with real data that was acquired from cryptocurrency exchanges.

Saleem et al. [15] have developed examining the effects of decrypting bitcoins on financial stability. It aims to give a thorough grasp of how cryptocurrencies affect and interact with the strength of the US dollar, inflation rates, stock market performance, and conventional banking operations. Granger causality testing, linear regression models, and case studies such as the Futures Exchange crash and Binance's successful integration were used to accomplish this.

d) Research Gap and Motivation

This section delves into the pertinent literature concerning the fundamentals of technical analysis alongside reinforcement learning methodologies. Additionally, it introduces the Proximal Policy Optimization (PPO) algorithm, employed for training agents, and elucidates its advantages. Using a local-global incentive system to improve agent performance on both an individual and group level, within a collaborative MAPPO framework, every agent specializes in trading one token from the portfolio. The Cross-Asset Attention Network is then used to process the obtained representations and produce asset ratings for portfolio management. The entire architecture undergoes training via PPO. Comparative evaluation of the PPO agent's strategy against other DRL algorithms, including DQN and Double DQN, and numerous extensively researched execution techniques, reveals its superior performance. Notably, recent advancements in DRL algorithms have demonstrated promising outcomes in addressing complex issues, such as formulating profitable trading strategies. The main goal of training a DRL agent is to maximise Profit and Loss (PNL) returns by using a unique reward mechanism. The aforementioned disadvantages serve as the driving force behind this study.

e) Contribution

The primary contributions are outlined below:

- At first, the data are gathered via the data of OHLCV market data.
- Affine-Mapping Based variational Ensemble Kalman Filter to clean the data at OHLCV market data.
- The pre-processed data are fed into the MFSGAN to effectively categorize the cryptocurrency trading levels as high risk, low risk and no risk.
- Using the proposed EQTS-CT-MFSGAN approach, performance metrics such as cumulative profit and loss, sensitivity, specificity, accuracy, and precision are examined.

f) Organization

The remains of the paper are structured as: Part 2 Proposed Methodology, Part 3 Result and discussion and Part 4 Conclusion.

II. PROPOSED METHODOLOGY

In this section, EQTS-CT-MFSGAN is proposed. Fig. 1 illustrates the proposed methodology's block diagram. This process consists of five steps: Data Acquisition, pre-processing, classification, and optimization. The process involves collecting data from OHLCV market data, followed by pre-processing using AM-VEnKF to clean the data. The pre-processed data is then input into the MFSGAN to classify cryptocurrency trading into high risk, low risk, and no risk categories. The HOA is utilized to optimize the weight parameter of the MFSGAN. As a result, a thorough explanation of each step is provided below.

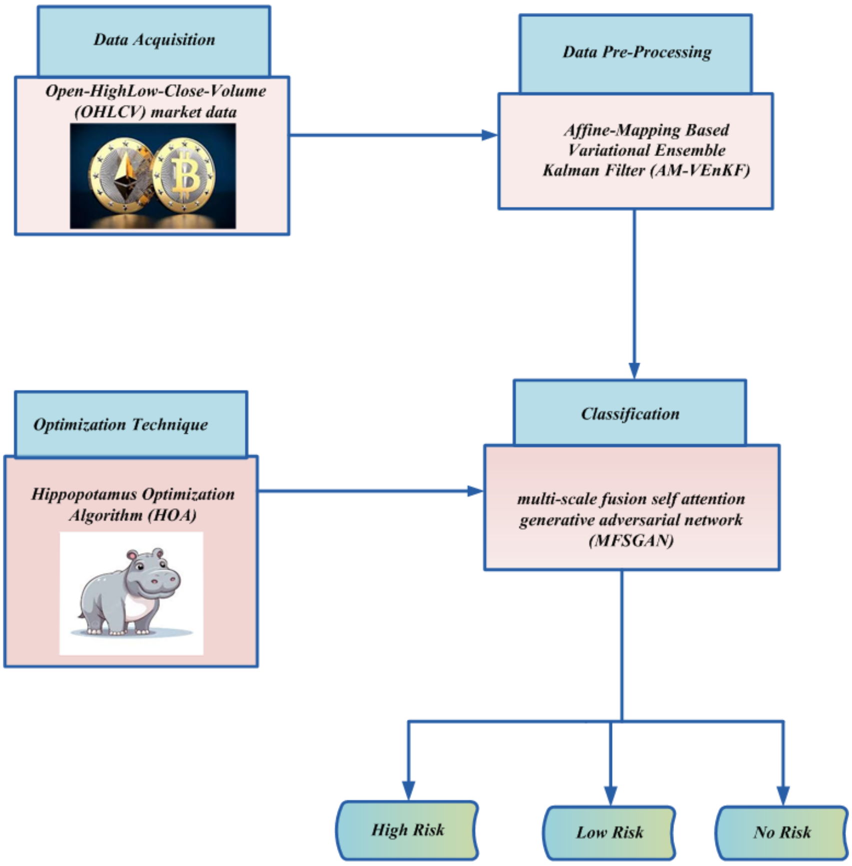

Fig 1: Proposed methodology's block diagram

A. Data Acquisition

In this study, OHLCV market data is employed [16]. This report includes five well-known cryptocurrency tokens that were downloaded using the CoinAPI platform. The files provide historical OHLCV data for November 2022 for Bitcoin, Ethereum, Cardano, Litecoin, and XRP3.

B. Preprocessing using Affine-Mapping Based variational Ensemble Kalman Filter (AM-VEnKF)

AM-VEnKF is a method used in data assimilation, particularly in the context of numerical weather prediction and other environmental modeling systems [17]. The goal is to compute the posterior distribution $\pi(xt / y_1 : t)$, assuming that the prior distribution $\pi(xt / y_1 : t-1)$ has been determined. Let's start with a brief summary of posterior distribution computation techniques based on transport maps. Creating a mapping that moves the prior distribution into the posterior is the key to these techniques. Assume $\tilde{x}t$ is distributed according to the previous $\pi(./ y_1 : t-1)$. The objective is to find a bijective mapping $T : X \rightarrow X$ such that $xt = T(\tilde{x}t)$ is distributed according to the posterior $\pi(./ y_1 : t)$. However, exact attainment of such a mapping is often impractical. In such scenarios, an approximate approach becomes necessary. To be more precise, let $\pi_T(\bullet)$ stand for the distribution of $xt = T(\tilde{x}t)$where $\tilde{x}t \approx \pi(.|y_1 : t-1)$. Finding a mapping $T \in H$ is the objective, where H is an assigned function space. Regarding a chosen distance metric that separates the 2 distributions, the goal of this mapping is to get $\pi_T(\bullet)$ as near as feasible to the real posterior, $\pi(./ y_1 : t)$. Actually, the definition of the KLD is:

$$D_{KL}(\pi_1, \pi_2) = \int \log\left[\frac{\pi_1(x)}{\pi_2(x)}\right]\pi_1(x)dx \tag{1}$$

Specifically, solve the following minimization problem to discover a mapping T.

$$\min_{T \in H} D_{KL}(\pi_T, \pi(x_t / y_1 : t)) \tag{2}$$

In actuality, the previous distribution $\pi(\tilde{x}t / y_1 : t-1)$ $x(x$ is typically not accessible analytically, and an ensemble of particles especially serves as their representation. As in the traditional EnKF, a Gaussian approximation of the ensemble's prior distribution $\pi(\tilde{x}t / y_1 : t-1)$ is estimated. To be more precise, take an assembly $\{\tilde{x}mt\}_m^M = 1$that is derived from the previous distribution $\pi(\tilde{x}t / y_1 : t-1)$, and then build an estimated prior $\pi(. / y_1 : t-1) = N\left(\tilde{\mu}t, \sum_t\right)$ using

$$\tilde{\mu}_t = \frac{1}{M}\sum_{m=1}^{M}\tilde{x}_t^m \tag{3}$$

$$\sum_t = \frac{1}{M-1}\sum_{m=1}^{M}\left(\tilde{x}_t^m\right) \tag{4}$$

Equation (2) is modified as,

$$\min_{T \in H} D_{KL}\left(\pi_T, \pi(x_t / y_1 : t)\right), \tilde{\pi}(. / y_{1:t})\alpha\tilde{\pi}(. / y_{1:1} - 1) \tag{5}$$

Specifically, aim to reduce the separation between π_T and the approximate posterior $\pi(x_t / y_1 : t)$ Finally, the data is pre-processed by AM-VEnKF, which cleans the data. These pre-processed data are fed into the classification using MFSGAN.

C. Classification using MFSGAN

In this section, classification MFSGAN is discussed [18]. The MFSGAN is used to classify the cryptorcurrency trading risk levels of high risk, low risk and no risk. Multi-scale fusion and attention mechanisms work together to give MFSGAN the ability to produce with better perceived quality. The model can produce visually appealing results that closely mimic genuine photos by capturing complex patterns, textures, and structures. Investors employ technical indicators to make trading strategy formulation easier and to simplify market information. Thus, the MFSGAN is given in equation (6)

$$D_{CON} = \sqrt{\left(L_{QT}^* - L_{BT}\right)^2 + \varepsilon^2} \tag{6}$$

Here, L_{QT}^* denotes the technical indicators, L_{BT} represents the discounted cumulative return, ε represents the penalty factor, D and signifies the loss function. The input level, first concealed level, second hidden level, and output level are the covers that are added to the method one after the other in a sequential fashion. The nerve cells in input and output covers of MFSGAN are determined by the amount of input-output in the feature data collection, whereas nerve cells in the inner covers are randomly or specifically computed using conventional criteria. The policy can employ a state value function, defined as (7), to calculate the expected return of a particular state D_e.

$$D_e = \sqrt{\left(D(L_N) - D(L_K)\right)^2 + \varepsilon^2} \tag{7}$$

Where, $D(L_N)$ and $D(L_K)$ denotes the state values are adequate to specify the ideal course of action, ε represents the penalty factor, D signifies the loss function. The input level, it is the initial level, contains 12 input nodules, each of which represents a sound spectrogram depends on the feature vector feature. The second level is hidden1. It is the first internal level, with 24 neurons in an optimal design that processes inputs using an activation function for Rectified Linear Units (ReLU). ReLU is the applied activation function since it works well with big size MFSGAN and solves the issue of disappearing gradients during trading risk. The market orders as like optimize a policy, policy gradien, and Locustella naevia is recognized by calculating the equation (8)

$$D_{hn} = \log\left(1 - P(B(z))\right) \tag{8}$$

Where, z represents original rain map, $B(z)$ low price volatility generated by production network, $P(B(z))$ represents the output of the discriminator. The third layer is unseen layer 2. It is the second internal cover, which employs ReLU activation function and comprises 24 optimized neurons. Every neuron in this layer has an associated weight and is coupled to every neuron's output in hidden layer 1. The dimensions of the hidden layer's joining weight matrix. The cryptocurrency markets such as safe investments, unprofitable actions and a social indicator is recognized by formulating the equation (9)

$$D_{nd} = -\log P(x) - \log(1 - P(B(z))) \tag{9}$$

Here, x signifies clean free data, z denotes original rain map, $P(B(z))$ signifies output of the discrimination, D signifies less function. The output layer, which comprises eight output nodules for every class to signifies a one-hot encoded mark linked to input features, is the fourth layer. The output layer employs the activation function of sigmoid. Every node in the output layer has a weight associated with it and is related to the outputs of every neuron in the concealed layer 2.The dimensions of the hidden layer's joining weight matrix. The market orders as such as risky is recognized by calculating the equation (10)

$$D_Q = D_{nd} + \beta \times D_{hn} \tag{10}$$

Where, D signifies the loss function, β denotes the weight of feature mapping, D_{nd} represents the large number of data, D_{hn} is the realizability of the network. The MFSGAN is trained and evaluated after its layered architecture is created. Greater safety-seeking investors could select larger window sizes, whilst more daring investors might select smaller window sizes. The size of the market orders feature matrix data is used by the model. Every output class label has a one-hot encoded form. Finally, cryptocurrency trading is identified as by using MFSGAN.Therefore, in order for the optimisation method to optimise the MFSGAN, the weight parameter ε^2, β is crucial. In this case, HOA is used to adjust the MFSGAN's weight and bias ε^2, β parameter.

D. Optimization using HOA

The HOA is used to optimize the weight parameter of MFSGAN. The fascinating hippopotamus is a member of the vertebrate class, specifically the mammal family. It is located in Africa. Semi-aquatic by nature, hippopotamuses predominantly inhabit aquatic environments such as rivers and ponds. Adults can remain submerged underwater for up to 5 minutes. Although they resemble shrew-like poisonous mammals, their closest cousins are whales and dolphins, with which they had a common ancestor around 55 million years ago. Hippopotamuses are herbivores who consume grass, branches, leaves, and other plant material as their primary food source, yet they may still be curious and investigate new food sources. Biologists caution against meat consumption, which can lead to digestive issues in these animals [19]. Hippopotamuses are among the most hazardous mammals because of their strong teeth, hostile personalities, and territorial tendencies.

Step 1: Initialization

Set the weight parameter values of generator ε^2, β from DCGNN to initialise the HOA population. The search agents of HOA, a population-based optimization method, are hippopotamuses. Hippos are potential solutions for the optimization issue in the HOA method, which means that each hippopotamus's position update values for the decision variables are represented in the search space. In this stage, the following formula is used to generate the vector of choice variables.

$$X_i ; x_{ij} = ab_j + r.(ab_j - ab_j), i = 1,2,......, N, j = 1,2......., m \tag{11}$$

Here, r denotes a arbitrary count between 0 and 1, X_i specifies the location of the i^{th} candidate solution, and ab and lb stand for the j^{th} decision variable's upper and lower bounds, respectively. Considering that the problem's decision variable count is denoted by m . and N represents the count of hippopotamuses in the herd.

Step 2: Random Generation

Using HOA method, the input fitness function acquired randomization after startup.

Step 3: Fitness function

The fitness function evaluation makes advantage of the weight parameter optimization ε^2, β effects. Equation (12) is where it is expressed.

$$Fitness\ function = optimizing[\varepsilon^2, \beta] \tag{12}$$

Step 4: Hippopotamus defense against predators for optimizing ε^2

The main reasons hippopotamuses live in herds are security and safety. These massive, heavily-weighted herds can discourage predators from getting too close. But because they are naturally curious, juvenile hippopotamuses can often wander out from the herd and end up as food for lions, spotted hyenas, and Nile crocodiles because they are not as strong as adult hippopotamuses. Hippocampal diseases can also make them susceptible to predators. Hippopotamuses' principal defence manoeuvre is quickly turning in the direction of the predator and making loud noises to scare it away. In order to effectively fend off the threat, hippopotamuses may approach the predator during this phase and cause it to withdraw. The location of the predator in the search space is represented by equation (13).

$$predator\,\varepsilon^2 = LB_i + \vec{R}_8 \left(UB_i - LB_i\right) \tag{13}$$

Here, $\vec{R}_8$ stands for a random vector with a range of zero to one. Limits of the i[th] choice variable, both lower and higher are indicated by the symbols LB and UB, where R is a random integer between 0 and 1.

Step 5: Hippopotamus escaping from the Predator for optimizing β

A hippopotamus will try to flee from a predator if it comes across a pack of predators or is unable to fend off the predator using its defensive manoeuvres. Since spotted lions and hyenas shy away from water, it will usually take sanctuary in the closest lake or pond. The goal of this plan is to locate a safe spot not far from its present location. Equations (14) and (15) illustrate how a random place is produced close to the hippopotamuses' present location in order to imitate this behaviour. The hippopotamus finds a safer spot close to its present location and modifies its position accordingly if the newly formed location increases the value of the cost function. Here, t denotes both the MaxIter and the current iteration.

$$LB_i^{local} = \frac{LB_i}{s} \tag{14}$$

$$UB_i^{local} = \frac{UB_i}{s} \tag{15}$$

$$xi^{hippoe} : yij^{hippoe} = y_{ij} + \beta . \left(LB_i^{local} + t_1 . \left(UB_i^{local} - LB_i^{local}\right)\right) \tag{16}$$

Here, r_1 is an arbitrary vector or integer and xi^{hippoe} is the hippocampal location that was examined to identify the nearest safe site, chosen at random from each of the three possible possibilities. In other words, the proposed method has a greater exploitation quality as a consequence of the examined situations (r), which result in a more appropriate local search.

Step 6: Update the Best Solution

Equations (17) and (18) specify the immature or male and female hippopotamuses' location update within the herd. Numerous mature female hippos, calves, numerous adult male hippos, and dominant male hippos make form hippopotamus herds. Here, f_i is objective function value.

$$x_i = \begin{cases} x_i^{mhippo} & f_i^{mhippo} < f_i \\ x_i & else \end{cases} \tag{17}$$

$$x_i = \begin{cases} x_i^{fbhippo} & f_i^{fbhippo} < f_i \\ x_i & else \end{cases} . \tag{18}$$

It improves the proposed algorithm's exploration process and results in a better global search.

Step 7: Termination

The process halts when the solution is deemed optimal; otherwise, it iterates back to Step 3 for fitness calculation and proceeds through the subsequent steps until the optimal solution is attained.

Combining the HOA for parameter optimization with the MFSGAN for classification presents a promising strategy with both strengths and challenges. HOA efficiently navigates complex solution spaces, ensuring optimal parameter tuning for the classification system. Meanwhile, MFSGAN enhances classification accuracy by capturing intricate patterns in the data. This collaborative approach results in a more accurate and adaptable classification model. However, managing the complexity of integrating HOA and MFSGAN and the computational resources required for training could pose hurdles. Overall, this approach holds great potential for improving classification accuracy but requires careful consideration of its complexities and resource demands.

III. RESULT AND DISCUSSION

This section discusses the experimental results of the proposed approach. Next, Python is used to simulate the proposed approach using the specified performance metrics. The proposed EQTS-CT-MFSGAN approach's results are examined using existing systems such as PNRCT-MAPPO, UCT-TCN, and OPCLO-DDQN, in that order.

A. Performance Measures

In order to choose the optimal classifier, this is an important task. Performance measures including cumulative profit and loss, sensitivity, specificity, accuracy, and precision are analysed to assess performance.

1) Accuracy

Accuracy is the capacity to measure an exact value. One statistic that may be used to quantify how well a model performs across all classes is accuracy. The following stated equation (19) is used to measure it.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{19}$$

In this step TP specifies True positive TN specifies True negative FP specifies false positive FN specifies false negative.

2) Precision

Equation (20) is provided for precision estimate, which includes several positive labels with high accuracy that was predicted.

$$\Pr ecision = \frac{TP}{(TP + FP)} \tag{20}$$

3) Sensitivity

Sensitivity usually refers to how accurately the graph displays small changes in data values. It also finds the proportion of positive samples and it is expressd equation (21)

$$Sensitivity = \frac{Tp}{Tp + Fn} \tag{21}$$

4) Specificity

Specificity is the proportion of real negatives that the approach accurately detects. The answer is found in equation (22),

$$Specificity = \frac{TN}{TN + FP} \tag{22}$$

B. Performance Analysis

The EQTS-CT-MFSGAN method's simulation outcomes are shown in Fig. 2 to 7. Then, the proposed CTDP-MMIM-DCGNN technique is likened with existing OPCLO-DDQN, UCT-TCN and PNRCT-MAPPO methods.

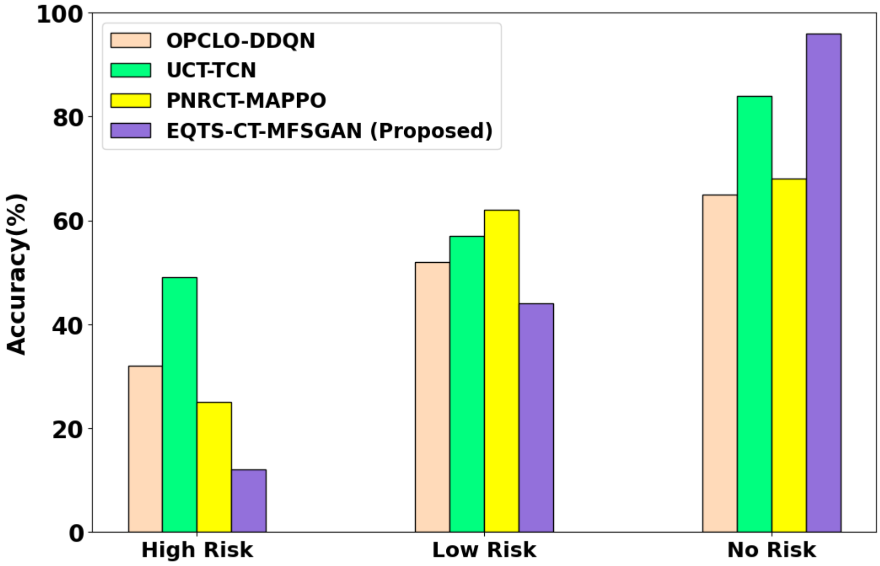

Fig 2: Performance analysis of Accuracy

Fig 2 depicts accuracy analysis. The EQTS-CT-MFSGAN attains 21.34%, 23.54% and 23.76% lower accuracy for high risk; 20.38%, 21.19% and 20.53% less accuracy for low risk; 23.76%, 21.65% and 21.34% higher accuracy for no risk; when evaluated to existing OPCLO-DDQN, UCT-TCN and PNRCT-MAPPO method.

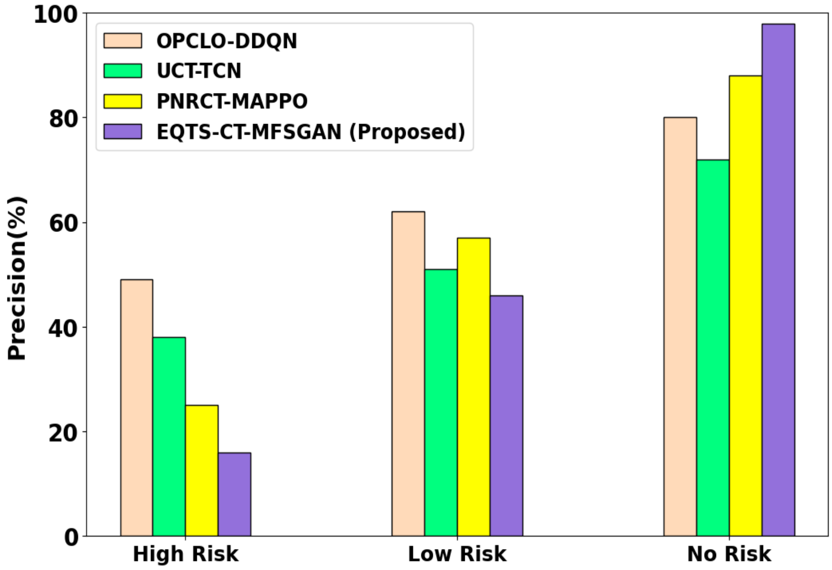

Fig 3: Performance analyses of precision

Fig 3 depicts precision analyses. The EQTS-CT-MFSGAN attains 21.34%, 21.39% and 20.28% lower precision for high risk; 19.28%, 20.42% and 20.63% less precision for low risk; 21.54%, 23.46% and 21.41% higher precision for no risk; when evaluated to existing OPCLO-DDQN, UCT-TCN and PNRCT-MAPPO method.

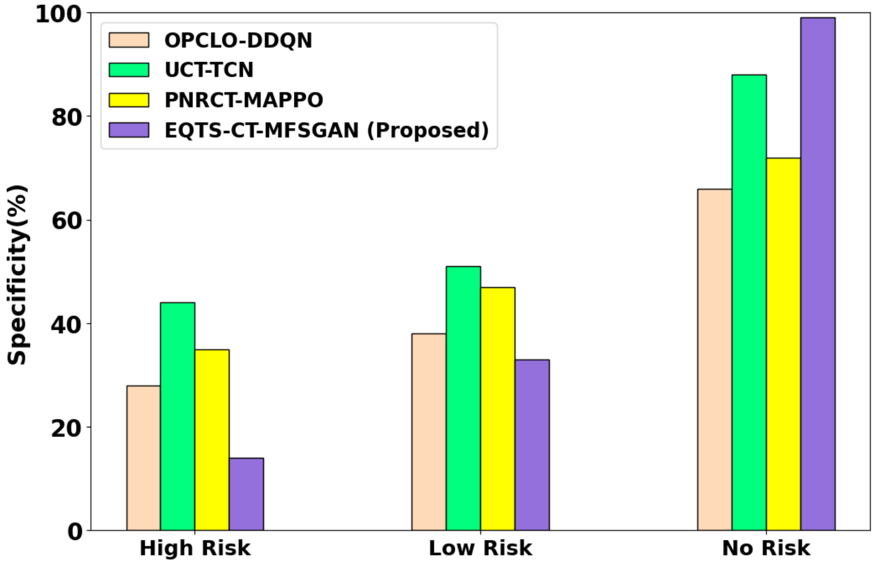

Fig 4: Performance analysis of specificity

Fig 4 depicts specificity analysis. The EQTS-CT-MFSGAN attains 21.38%, 21.29% and 23.92% lower Specificity for high risk; 22.94%, 23.29% and 21.49% less specificity for low risk; 21.32%, 23.51% and 23.35% higher specificity for low risk; when evaluated to existing OPCLO-DDQN, UCT-TCN and PNRCT-MAPPO method.

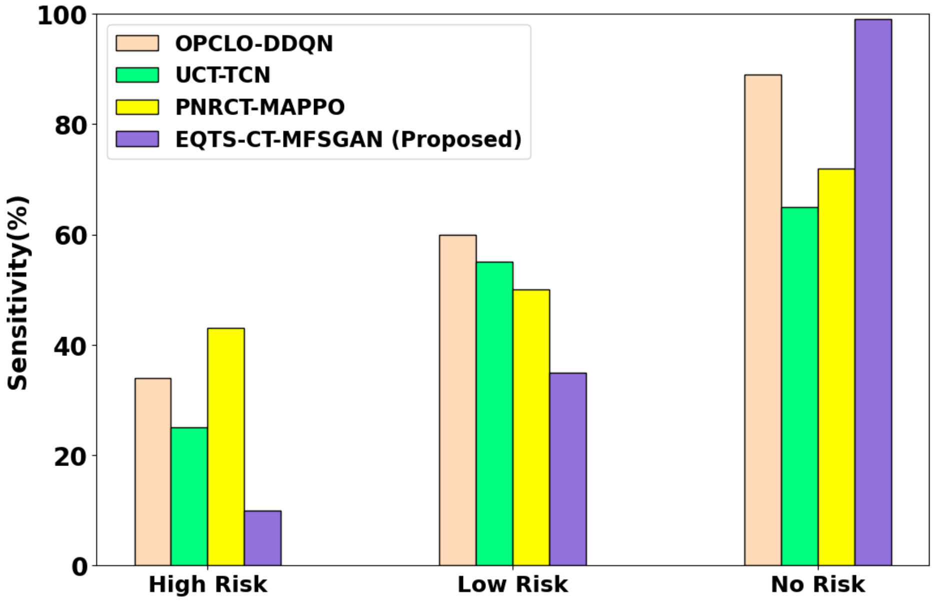

Fig 5: Performance analyses of sensitivity

Fig 5 depicts sensitivity analyses. The EQTS-CT-MFSGAN attains 18.82%, 20.53% and 23.79% lower sensitivity for high risk; 23.46%, 23.58% and 21.52% less sensitivity for low risk; 24.31%, 23.19% and 24.25% higher sensitivity for no risk; when evaluated to existing OPCLO-DDQN, UCT-TCN and PNRCT-MAPPO method.

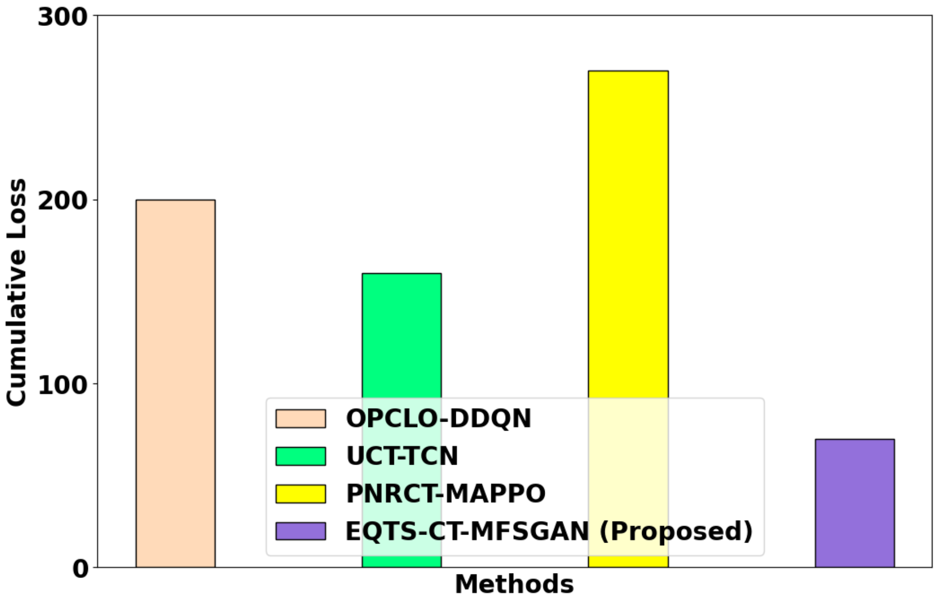

Fig 6: Performance analysis of cumulative loss

Fig 6 depicts the cumulative loss analysis. The EQTS-CT-MFSGAN attains 24.51%, 23.67% and 23.87% lower cumulative loss; when evaluated to existing OPCLO-DDQN, UCT-TCN and PNRCT-MAPPO method.

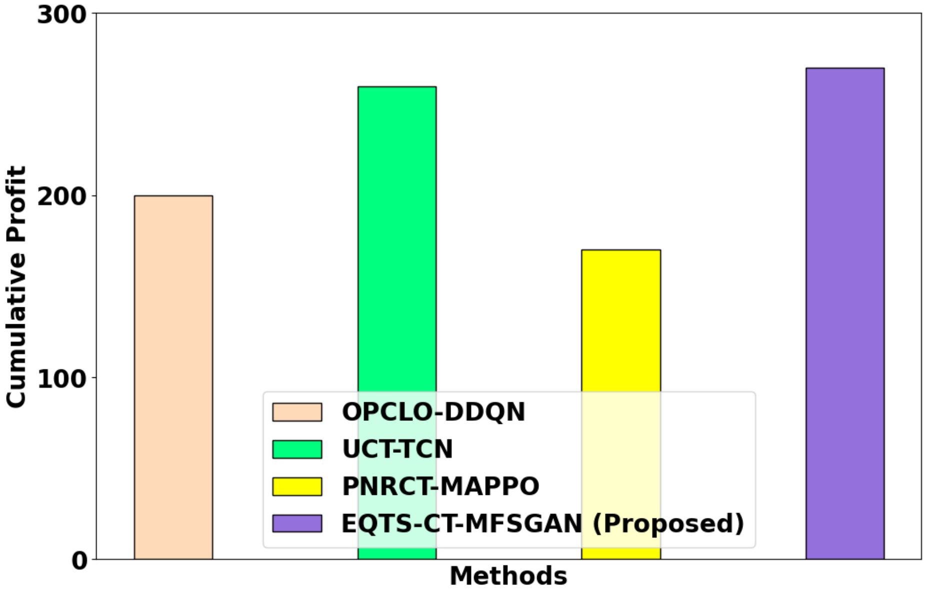

Fig 7: Performance analysis of cumulative profit

Fig 7 depicts the cumulative profit analysis. The EQTS-CT-MFSGAN attains 21.38%, 21.29% and 23.92% lower cumulative profit; when evaluated to existing OPCLO-DDQN, UCT-TCN and PNRCT-MAPPO method.

C. Discussion

An EQTS-CT-MFSGAN model for an OHLCV market data is developed in this paper. The EQTS-CT-MFSGAN method involves encompasses based data pre-processing. The approach's highest average results were compared to the average results for existing approaches such as OPCLO-DDQN, UCT-TCN, and PNRCT-MAPPO, using market data from the OHLCV instance. In terms of cumulative profit, EQTS-CT-MFSGAN achieves reductions of 21.38%, 21.29%, and 23.92% compared to existing methods respectively. Conversely, in cumulative loss, EQTS-CT-MFSGAN demonstrates improvements, with reductions of 24.51%, 23.67%, and 23.87% relative to the same methods. Sensitivity analysis unveils EQTS-CT-MFSGAN's lower sensitivity for high and low risk by 18.82% to 24.31%, and 20.53% to 23.58% respectively, yet higher sensitivity for no risk by 23.19% to 24.25%. Specificity analysis further emphasizes these differences; with EQTS-CT-MFSGAN showcasing reductions in specificity for high and low risk by 21.38% to 22.94%, and 21.29% to 23.29% respectively, while exhibiting increased specificity for no risk by 21.32% to 23.51%. The proposed method CTDP-MMIM-DCGNN has high specificity and accuracy evaluation metrics for no risk than existing methods. These findings underscore the nuanced performance trade-offs and highlight EQTS-CT-MFSGAN's potential strengths and weaknesses in managing various risk scenarios compared to established methodologies.

IV. CONCLUSION

In conclusion, this manuscript presents an exploration of quantitative trading strategy of reinforcement learning in cryptocurrency trading. During pre-processing, the data is cleaned using the Affine-Mapping Based Variational Ensemble Kalman Filter. The pre-processing result is forwarded to the MFSGAN is to efficiently classifies the risk levels of high risk, low risk and no risk. The proposed EQTS-CT-MFSGAN approach is implemented in Python utilization of Open-HighLow-Close-Volume (OHLCV) market data. The proposed approach is examined in a variety of scenarios, including those involving computing time, cumulative profit and loss, sensitivity, specificity, accuracy, and precision. Presentation of proposed EQTS-CT-MFSGAN method covers 30.56%, 21.76%, 35.97% higher specificity for highly risk; and 29.47%, 38.76% and 28.78% lower computational time for no risk analyzed to the existing methods such as OPCLO-DDQN, UCT-TCN and PNRCT-MAPPO respectively. The exploration of quantitative trading strategies using reinforcement learning in cryptocurrency markets could be extended in several directions. Furthermore, by making state space simpler, a feature focus on technical indicators may further enhance TraderNet performance. Limitations of the current study include data constraints, model complexity, risk of overfitting, assumptions of market stationarity, and regulatory/compliance risks.

Acknowledgement

1. Funded by Sichuan Urban-Rural Integration Research Center, Key Research Base of Social Sciences of Mianyang City, Project name: "Research on the impact of Industrial Agglomeration on Urban-Rural Integration Development of Chengdu-Chongqing Economic Circle" (Project number: SCCXRH2023ZC03);
2. 2024 Bazhong City Social Science Planning Project: Research on Bazhong City's overall Marketing and City Reputation and Influence under the background of high-quality Development (Project number: BZ24YB154);
3. Social Science Planning Project of Bazhong City in 2024: Research on the synergistic impact of scientific and technological innovation and Industrial structure Optimization on the high-quality economic development of Bazhong City and the improvement path (Project number: BZ24YB043**).**

REFERENCES

[1] Delfabbro, P., King, D. L., & Williams, J. (2021). The psychology of cryptocurrency trading: Risk and protective factors. *Journal of behavioral addictions*, *10*(2), 201-207.

[2] Delfabbro, P., King, D., Williams, J., & Georgiou, N. (2021). Cryptocurrency trading, gambling and problem gambling. *Addictive Behaviors*, *122*, 107021.

[3] Sattarov, O., Muminov, A., Lee, C. W., Kang, H. K., Oh, R., Ahn, J., ... & Jeon, H. S. (2020). Recommending cryptocurrency trading points with deep reinforcement learning approach. *Applied Sciences*, *10*(4), 1506.

[4] Liu, Z., & Zhang, R. (2024). An Empirical Study on Herd Behavior in Cryptocurrency Trading. *Journal of Computer Information Systems*, *64*(3), 438-452.

[5] Jing, L., & Kang, Y. (2024). Automated cryptocurrency trading approach using ensemble deep reinforcement learning: Learn to understand candlesticks. *Expert Systems with Applications*, *237*, 121373.

[6] McGivern, P. (2023). A broader perspective on cryptocurrency trading: consumer-driven value, online communities and heuristics are drivers for consumer behaviour. *Journal of Money and Business*, *3*(2), 137-146.

[7] Park, S., & Yang, J. S. (2023). Intelligent cryptocurrency trading system using integrated AdaBoost-LSTM with market turbulence knowledge. *Applied Soft Computing*, *145*, 110568.

[8] Johnson, B., Stjepanović, D., Leung, J., Sun, T., & Chan, G. C. (2023). Cryptocurrency trading, mental health and addiction: A qualitative analysis of reddit discussions. *Addiction Research & Theory*, *31*(5), 345-351.

[9] Schnaubelt, M. (2022). Deep reinforcement learning for the optimal placement of cryptocurrency limit orders. *European Journal of Operational Research*, *296*(3), 993-1006.

[10] Kochliaridis, V., Papadopoulou, A., & Vlahavas, I. (2024). UNSURE-A machine learning approach to cryptocurrency trading. *Applied Intelligence*, 1-23.

[11] Kumlungmak, K., & Vateekul, P. (2023). Multi-Agent Deep Reinforcement Learning With Progressive Negative Reward for Cryptocurrency Trading. *IEEE Access*.

[12] Su, Z. (2024). Analysis of Cryptocurrency and Its Combination with Quantitative Transactions. *In SHS Web of Conferences* (Vol. 181, p. 02016). EDP Sciences.

[13] Yu, H., Sun, Y., Liu, Y., & Zhang, L. (2024, March). Bitcoin Gold, Litecoin Silver: An Introduction to Cryptocurrency Valuation and Trading Strategy. *In Future of Information and Communication Conference* (pp. 573-586). Cham: Springer Nature Switzerland.

[14] Fang, F., Chung, W., Ventre, C., Basios, M., Kanthan, L., Li, L., & Wu, F. (2024). Ascertaining price formation in cryptocurrency markets with machine learning. *The European Journal of Finance*, *30*(1), 78-100.

[15] Saleem, M. N., Doumenis, Y., Katsikas, E., Izadi, J., & Koufopoulos, D. (2024). Decrypting Cryptocurrencies: An Exploration of the Impact on Financial Stability. *Journal of Risk and Financial Management*, *17*(5), 186.

[16] Kochliaridis, V., Kouloumpris, E., & Vlahavas, I. (2023). Combining deep reinforcement learning with technical analysis and trend monitoring on cryptocurrency markets. *Neural Computing and Applications*, *35*(29), 21445-21462.

[17] Wen, L., & Li, J. (2022). Affine-mapping based variational ensemble Kalman filter. *Statistics and Computing*, *32*(6), 97.

[18] Xue, W., Huan-Xin, C., Sheng-Yi, S., Ze-Qin, J., Kai, C., & Li, C. (2022). MSFSA-GAN: Multi-Scale Fusion Self Attention Generative Adversarial Network for Single Image Deraining. *IEEE Access*, *10*, 34442-34448.

[19] Amiri, M. H., Mehrabi Hashjin, N., Montazeri, M., Mirjalili, S., & Khodadadi, N. (2024). Hippopotamus optimization algorithm: a novel nature-inspired optimization algorithm. *Scientific Reports*, *14*(1), 5032.