Quantitative Finance and Economics

QFE, 9(3): 658–681.
DOI: 10.3934/QFE.2025023
Received: 05 April 2025
Revised: 26 May 2025
Accepted: 02 July 2025
Published: 08 September 2025

https://www.aimspress.com/journal/QFE

Research article

Cryptocurrency price prediction and portfolio optimization based on LSTM and multi-task neural network

Weiqi Chen[1] and Hang Zheng[2,*]

[1] School of Communication and Information Engineering, Shanghai University, Shanghai, China
[2] Faculty of Science, The Chinese University of Hong Kong, Hong Kong, China

* **Correspondence:** Email: 1155215120@link.cuhk.edu.hk.

Abstract: The precise forecast of cryptocurrency prices is essential for portfolio investment because of their volatility and operability in virtual trading markets, which poses a huge trouble to investors' decision-making ability and investment planning. In this paper, we concentrated on the prediction of the recent trends of mainstream cryptocurrencies and selected them to optimize the portfolio to maximize profit and reduce risk. We used neural networks to solve this problem, which has three layers, including the LSTM layer, dropout layer, and dense layer. We focused on BCH, BTC, ETH, ETC, LTC, EOS, and XRP and collected their datasets for estimations. An LSTM model, a multi-task learning model, and a novel loss function, where the Negative Sharpe Ratio is provided, were implemented to predict the best portfolio (weights) for the cryptocurrencies mentioned above. The common evaluation indices, such as MSE, RMSE, MAE, and R-square (R^2), can demonstrate the accuracy and reliability of the Neural Network models. Due to the significant price differences among currencies, the values of MSE, MSE, and MAE were large, making it difficult to evaluate their accuracy. Therefore, R^2 was adopted. Finally, we simulated the portfolio investment and saw the revenue compared to the existing approaches. The new machine learning model abandoned the previous methods, which only predicted and analyzed a single cryptocurrency and ignored the correlations among them. Therefore, our innovation of this research was to use neural networks to consider investment plans combining multiple currencies while minimizing volatility and ensuring a large Sharpe Ratio, thereby obtaining the best portfolio investment of cryptocurrencies.

Keywords: optimization; machine learning; neural network; cryptocurrency forecast; portfolio investment; risk analysis

JEL Codes: G11, G17, C51, C58

1. Introduction

Cryptocurrency markets are full of uncertainty and volatility, but returns are also considerable compared to other investments. Therefore, it has attracted the attention of many investors and researchers. Recently, it has become a financial asset and created astonishing price appreciation and fluctuation (Corbet et al., 2019).

The prediction and analysis of cryptocurrency price fluctuations is an important hot topic at the moment. The reason is that the price of cryptocurrency changes very rapidly, sometimes $10,000 oscillations in a day, and it is tough to catch the turning point and reset the portfolio we made. Hence, it is crucial to invent an efficient model that can recognize the potential price change from historical data and then re-estimate the investment combination that we did before. The model should reduce risks, help better determine decision making, and create as much profit as possible.

Traditional cryptocurrency price forecasting tries to focus on single cryptocurrency prediction by using machine learning methods, and they struggle to capture every detail of the price wave visually. For instance, the price is influenced by many factors, such as exchange rate, gold prices, and alternative coins (Angela & Sun, 2020). Consequently, these models promote the application and update of machine learning algorithms in cryptocurrency, providing new perspectives and tools for researchers and investors. In this paper, we draw on the machine learning, deep learning, and neural network algorithms in past papers, but through optimization and updates, make the current cryptocurrency price prediction involving investment portfolios more in-depth and accurate. We use historical data of OKX's official mainstream cryptocurrencies and other necessary price features, such as closing price, volatility, and trading volume, to develop a better model to capture the price levels of each cryptocurrency and its investment portfolio. We successfully handle the complex and ever-changing currency price fluctuation analysis by combining LSTM and multi-task learning algorithms, along with a backpropagation neural network. Furthermore, after we capture the price features for each coin, we incorporate their characteristics and simulate an investment combining all those mainstream coins to look at the profit that we can make through this new method. This is also the best way to test the portfolio algorithm and verify its efficiency.

A general prospect of this paper is to discover the detailed feature through the dataset of mainstream cryptocurrencies and not only analyze single coin change but also help investors make decisions with the portfolio, which is more practical and safer in real life. We ensure the authenticity and clarity of turning points in each category to better construct the model's reliability, utility, and accuracy. Moreover, we provide a systematic solution from coin price predictions to portfolio distribution, which can have a huge impact on theoretical research and investment options. By showing limitations in existing forecasting ways, we are dedicated to developing cryptocurrency portfolio decision making and providing perspectives for investors, researchers, and the general public in the virtual financial market.

The major contributions of this paper are:

Based on selecting the LSTM model and optimizing it to a certain extent, we achieve good results in predicting each cryptocurrency. Furthermore, to address the issue of only considering a single currency in current cryptocurrency research and the subjective factors in mainstream investment

decisions, a new model is proposed that utilizes deep learning and multi task learning neural networks (DLM) to consider the characteristics of multiple currencies while taking into account prediction bias. Based on these characteristics, an investment portfolio with acceptable volatility is calculated while maximizing profit margins.

2. Literature review

Many researchers have created conventional machine learning algorithms and models to forecast the changes and regular patterns of the cryptocurrency prices. In one survey (Khedr et al., 2021), it was concluded that Random Forest (RF), Recurrent Neural Network (RNN), Multilayer Perception (MLP), etc. are used in predicting the price of cryptocurrency. Additionally, other researchers have applied the moving average method to forecast the price change of Bitcoin Cash (Abu Bakar et al., 2019), using Probabilistic deep learning, transfer learning techniques, and Gated Recurrent Unit (GRU), to predict the price movement of Bitcoin (Amin Golnari et al., 2024). Moreover, some classification methods are also implemented to better discover the patterns of price changes in cryptocurrency. For instance, Support Vector Machine was used to make classifications on different characters that were shown in the Bitcoin price, and they were collected in a table to detect the price change factors and made better predictions (Zhao et al., 2019). The author made a return of up to 22.7% after trading fees.

Traditional time series models like Autoregressive Integrated Moving Average (ARIMA) outperformed other models in terms of mean square error (MSE), mean absolute error (MAE), and root mean absolute error (RMSE) in the prediction of Bitcoin, XRP, and Ethereum (Alahmari, 2019). Moreover, Econometrics time series models, such as Auto-Regressive Conditional Heteroskedasticity (ARCH) (Bollerslev, 2008), and Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) model (not constant variance) (Ferland et al., 2006), are used over time to simulate fluctuations in cryptocurrency prices. However, a traditional time series model cannot precisely capture the historical data in prediction. Studies have also shown that simple neural networks such as MLP are superior to complex models in predicting cryptocurrency volatility, especially exhibiting strong adaptability in asset allocation during high volatility periods such as epidemics (García-Medina & Luu Duc Huynh, 2021). Therefore, time series classification based on a recurrent long-short-term memory (LSTM) is also employed as an important approach to predict the price of cryptocurrency. Additionally, LSTM, CNN, Neural Network, and Gradient-Boosted Trees (GBTs) models were used to achieve the goal of improving the accuracy of price prediction. The researchers combined an LSTM model for the binary classification of cryptocurrency price trends. The $f1$ was 0.68 for the LSTM time series model, which is the highest among the testing models (Kwon et al., 2019).

Deep learning algorithms are also widely used in cryptocurrency price forecasting. LSTM and GRU have been popular methods for researchers to implement and compare. For instance, a hybrid LSTM-GRU model was displayed to construct the system's deep learning model after Min_max normalization (Patel et al., 2020). Besides, a comparative research of Bitcoin combined DNN, LSTM, CNN, and multilayer perception (MLP) with SVR class for regression and SVC for classification to improve the performance of price prediction accuracy (Ji et al., 2019).

On the other hand, some researchers have used transfer entropy to screen key driving factors and construct classification models, and found that Bitcoin has self-regulation during periods of high volatility and can maintain prediction accuracy without external economic variables (García-Medina & Aguayo-Moreno, 2024). However, with the gradual development and growth of the cryptocurrency

market, many other coins besides Bitcoin have become popular. For different cryptocurrency portfolios, the decision making becomes different. Diversification among cryptocurrencies has greatly improved the utility and Sharpe ratio (Liu, 2018). Hence, it is significant to not only consider single cryptocurrency price fluctuation but also think about how to optimize portfolios, according to the information in the single coin price forecast. $Log-returns$ has been used to estimate the best portfolio among Bitcoin, Ethereum, Ripple, Bitcoin Cash, and Litecoin, and the Sharpe ratio has been chosen to balance the profit and risk; the return of their best choice is 38.31% with a Sharpe ratio of 1.79 (Ma et al., 2020). Apart from this, researchers also use an asset-allocation model to set equally weighted portfolio with optimal of mean-variance, and finally got a $maximum-diversification$ model, which reaches 22% return (Petukhina et al., 2021).

In fact, many researchers have focused on studying the price change patterns of a single currency because they are easier. However, with the development of machine learning algorithms and the gradual combination of operations research algorithms and virtual currencies, more and more research has begun to move towards finding the best currency portfolio investment.

3. Methodology and architecture

3.1. Data source

In this study, we collected the cryptocurrency data from the OKX exchange. Initially, we planned to obtain data for 12 cryptocurrencies; however, only some cryptocurrencies had data coverage time that met the requirement of 1000 days. Therefore, to ensure a reasonable amount of cryptocurrency data and to draw robust and meaningful conclusions for the final results, we adopted cryptocurrencies that met only the 1000 days requirement. We obtained transaction data from the official API provided by OKX. Closing prices and trading volumes are commonly used as inputs for prediction models, and have achieved good results. Thus, our initial data covered the period from May 29, 2022 to February 22, 2025 (one per day), the currencies included BCH, BTC, EOS, ETC, ETH, LTC, and XRP (all denominated in US dollars), and the data included closing price and trading volume. Considering the significant price differences among cryptocurrencies, logarithmic scales were uniformly used for plotting. Figures 1 and 2 show the selected cryptocurrencies' daily closing price and trading volume.

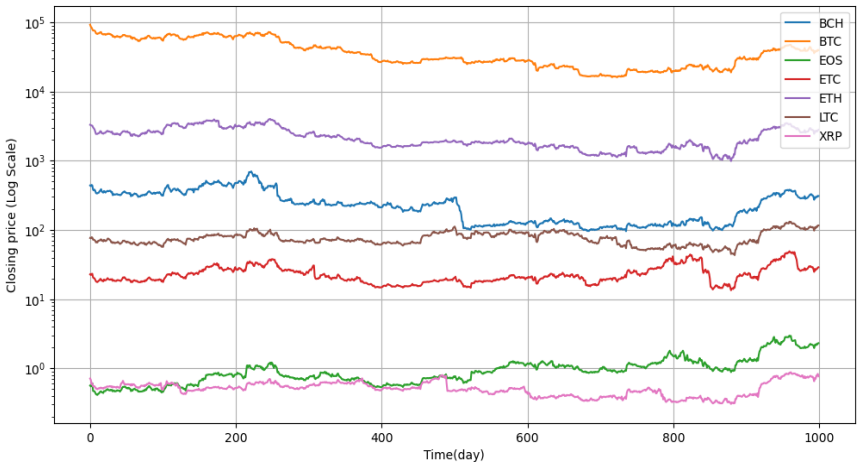

Figure 1. Daily closing price (Log Scale) for the selected cryptocurrency.

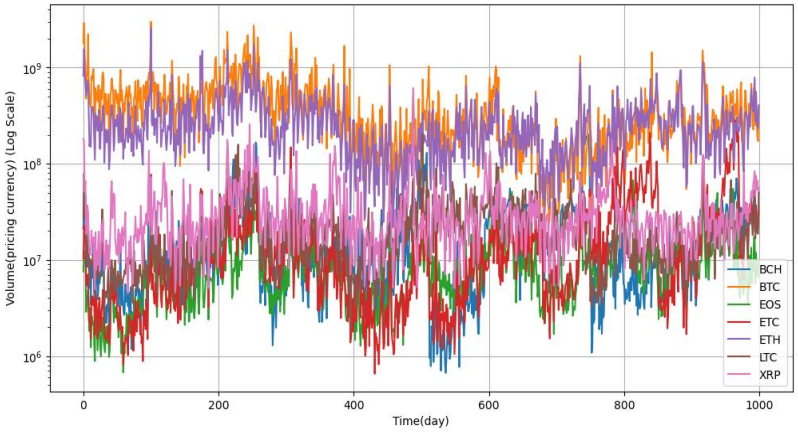

Figure 2. Daily volume (pricing currency) (Log Scale) for the selected cryptocurrency.

Correspondingly, some cryptocurrencies exhibit significant volatility. On one hand, this requires extra attention when optimizing investment portfolios to ensure acceptable volatility and returns. On the other hand, this also requires special attention to the volatility of cryptocurrencies when predicting prices, in order to optimize the prediction effect reasonably. Therefore, we also calculated the volatility of cryptocurrencies, as shown in Figure 3.

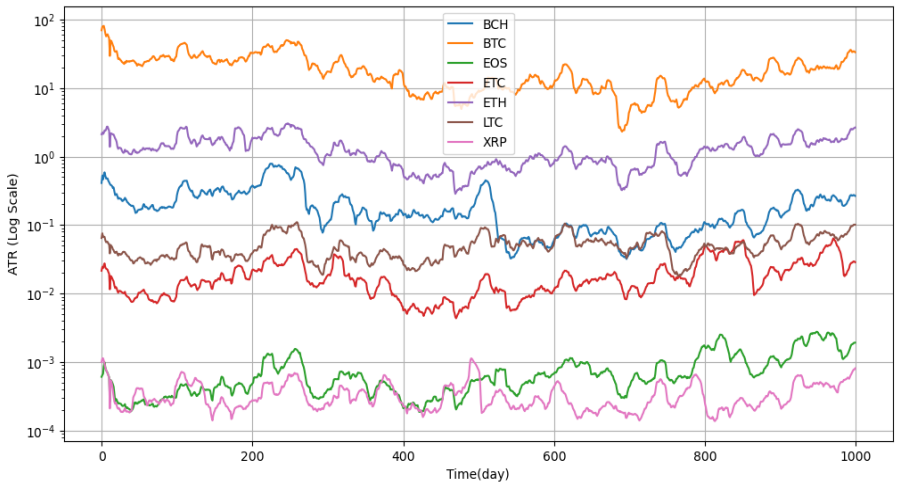

Figure 3. Daily ATR (Log Scale) for the selected cryptocurrency.

In the study (Cohen, 2022), attempts were made to design, optimize, and use average true range (ATR) based trading systems for five popular cryptocurrencies. It was proven that ATR-based systems could predict the price trend of the studied cryptocurrency. The introduction of ATR could optimize the accuracy of our model predictions to a certain extent. The volatility index was measured using ATR, and its formula is as follows:

$$ATR_t = \frac{1}{n}\sum_{i=1}^{n} TR_i \tag{1}$$

Among them,
N: Represents a smooth cycle (here it is 14 days)
ATR_t: Represents the ATR value of the t-th day
TR_i: Represents the TR value of the i-th day. TR_i is calculated as:

$$TR_i = \max\left(\begin{cases} High_i - Low_i \\ |High_i - Close_{i-1}| \\ |Low_i - Close_{i-1}| \end{cases}\right) \tag{2}$$

Among them,
$High_i$: Represents the highest price on the i-th day
Low_i: Represents the lowest price on the i-th day
$Close_i$: Represents the closing price on the i-th day
Specifically, for cases where $t < 14$:

$$ATR_t = \frac{1}{t}\sum_{i=1}^{t} TR_i \tag{3}$$

Table 1. Proportion of market value of selected cryptocurrencies.

Rank	Cryptocurrency Name	Market Cap	Rate
1	BTC	1.673 trillion	0.606
2	ETH	239.8 billion	0.0869
3	XRP	139.4 billion	0.0505
4	LTC	6.889 billion	0.00250
5	BCH	6.450 billion	0.00234
6	ETC	2.701 billion	0.000979
7	EOS	853.2 million	0.000309
8	ALL	2.069 trillion	0.750

As shown in Table 1, the seven cryptocurrencies used in this study account for 75% of the total market value. The market value data was sourced from https://coinmarketcap.com/. This indicated that the sample taken could represent the overall cryptocurrency market, and correspondingly, the results of the model speculation could also be more informative.

3.2. Algorithms

a) Overall Process

Features extracted from deep learning models are considered to perform better than traditional manually computed features (Zhang et al., 2020). The general process is shown in Figure 4. CNN and LSTM were used to predict the price of cryptocurrencies, and the final result of LSTM showed the best performance and the lowest error rate (Wen & Ling, 2023). LSTM was successfully applied in the research to predict multiple currencies (Malsa et al., 2021). Moreover, traditional LSTM models and AR-based LSTM models were used to successfully predict the price of Bitcoin (Wu et al., 2018). These results indicated that the LSTM model could predict the price of cryptocurrencies very well. Another

review article (Ruder, 2017) presents the overall architecture of Multi-Task Learning (MTL) and commonly used MTL methods. In this study, MTL enabled neural networks to capture hidden features in cryptocurrencies through data sharing.

First, the collected data of seven cryptocurrencies were cleaned and processed, and then predicted daily prices were obtained through LSTM model processing. Then, the daily price and the initial actual price were fused and processed entered into a multi-task neural network to obtain the final score for each cryptocurrency. Next, we used the Softmax function to convert the final rating of each cryptocurrency into the corresponding investment weights of the investment portfolio Finally, we conducted an investment simulation.

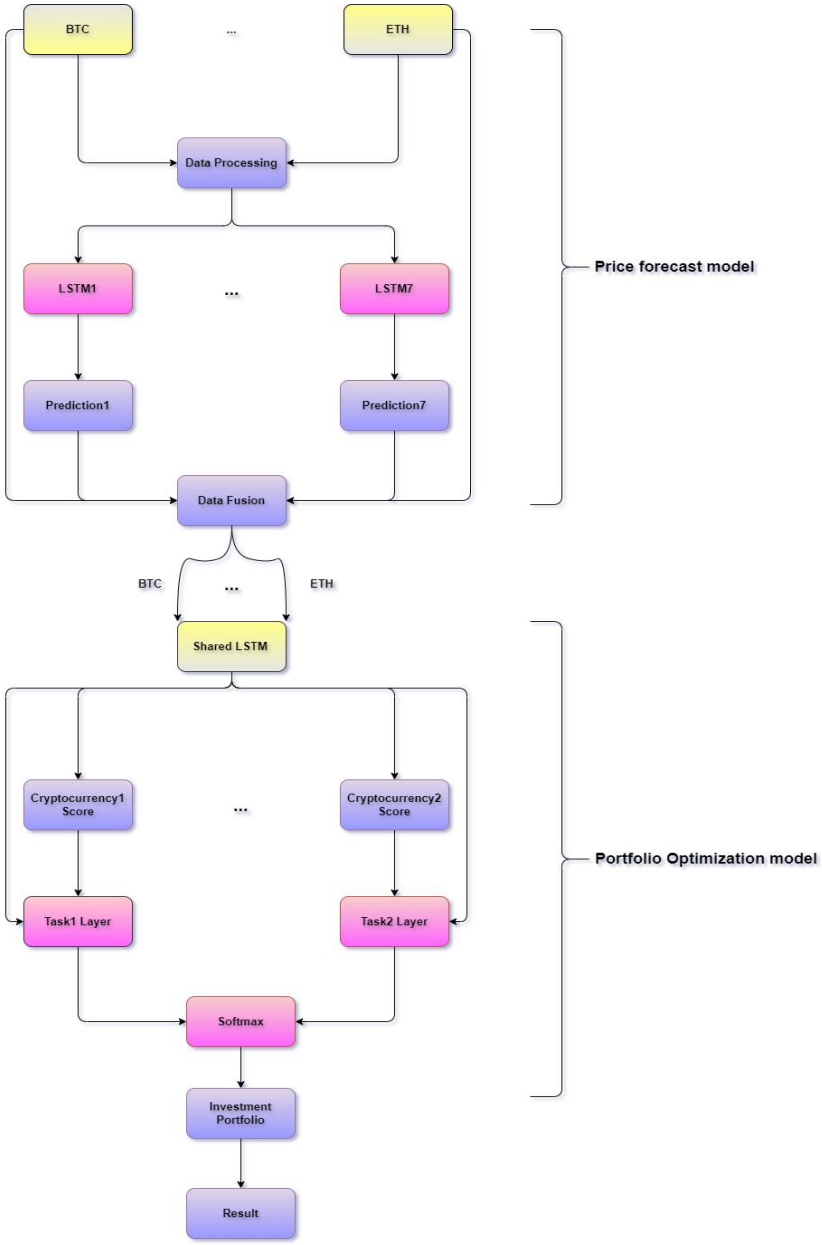

Figure 4. The general process.

b) Price Forecast

i) Data Processing

We planned to train a unique LSTM model for each cryptocurrency. Thus, we took each cryptocurrency's closing price, trading volume, and ATR as inputs to the model. We split the original data into training and testing sets using α as the segmentation ratio and divided the total N pieces of data based on window size M to obtain N-M+1 windows. The first M-1 values of each window were used as variable x, and the last was used as variable y. Finally, for each window, the first value $\boldsymbol{x_1}$ was used as the reference, and each subsequent value became the relative change $\boldsymbol{x_{norm-p}}$ of the first value, which was used for normalization. The formula is as follows:

$$x_{norm-p} = \frac{x_p}{x_1} - 1 \tag{4}$$

Among them:

$\boldsymbol{x_{norm-p}}$: Represents the p-th normalized value

$\boldsymbol{x_p}$: Represents the p-th value

ii) Model Architecture

Considering that the price of cryptocurrencies has a certain time relationship, it is advisable to consider their data as a time series. LSTM usually performs well in longer time series. Thus, we used the LSTM model to predict stock prices. The LSTM model is a special type of RNN (Recurrent Neural Network) designed to solve long sequence dependency problems, which can effectively alleviate the problems of gradient vanishing and exploding. Its general structure is shown in Figure 5.

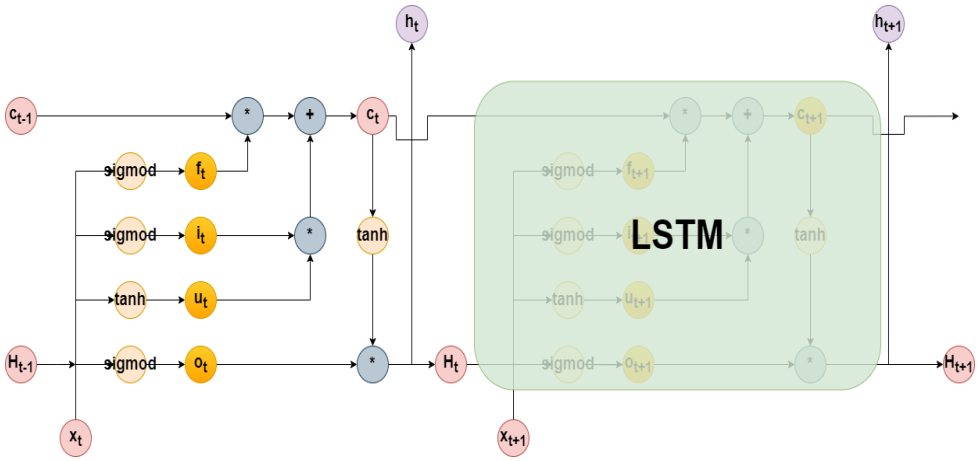

Figure 5. The LSTM general structure.

At each time step t, the calculation process of LSTM is as follows:
The first step is to calculate the forget gate

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{5}$$

Among them,
f_t: Between 0 and 1, close to 0 indicates forgetting, close to 1 indicates retention.
W_f, b_f: The weight and bias of the forget gate.
σ: Activation function.
The second step is to calculate the input gate

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{6}$$

$$u_t = \tanh(W_C * [h_{t-1}, x_t] + b_C) \tag{7}$$

Among them,
i_t: Control the amount of new information.
u_t: It is a candidate state, which is the new information calculated at the current time.
The third step is to calculate the cell state

$$C_t = f_t * C_{t-1} + i_t * u_t \tag{8}$$

Among them,
$f_t * C_{t-1}$: The forgotten part of the information.
$i_t * u_t$:Newly added information.
The last step is to calculate the output gate.

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{9}$$

$$h_t = o_t * \tanh(C_t) \tag{10}$$

Among them,
o_t: Control the amount of output information.
h_t: As the output of the current moment, it will also be passed on to the next time step.

Overall, it can more effectively capture long-term dependencies through gating mechanisms and has strong memory capabilities, enabling it to remember important historical information and ignore relatively irrelevant information.

For the prediction model in this experiment, the structure is shown in Figure 6. Initially, there is the input layer. Features represent the number of input features (closing price, trading volume, ATR), and Time Step represents the number of input historical data (here, predicting the 60th day based on the data from the previous 59 days). This input layer is used to clarify the input format of the data.

Next is the LSTM Layer, where Neurons represents the number of neurons (here, there are 100 neurons, generating 100-dimensional feature values), and Return Sequence represents whether to return a sequence (since subsequent time series processing is no longer required, the final value can be returned directly). This LSTM Layer is used to extract time series features, enabling the model to understand price change characteristics and better predict prices.

Next is the Dropout Layer, where Dropout Rate represents the random inactivation rate (0.2 in this case, meaning that there will be 20 random neurons that no longer work in the next layer). This layer is designed to make each neuron in the model more independent, enhance robustness, and reduce overfitting.

Next is the Dense Layer, where Neurons represents the number of neurons (here, there is only one neuron to ensure that the output only has one current price value), and Activation is the activation function (which can be analogized to the activation of human neurons). This layer is used to convert the obtained feature data into the final predicted price.

Finally, there is the Output Layer, which represents the number of output features (predicted prices) and is used to specify the data output form. There are three adjustable parameters for parameter selection: Time Step, Neurons, and Dropout Rate. We used the grid search method, with a Time Step size of 10, Neurons step size of 10, and Dropout Rate step size of 0.05, to gradually search and find the best parameter.

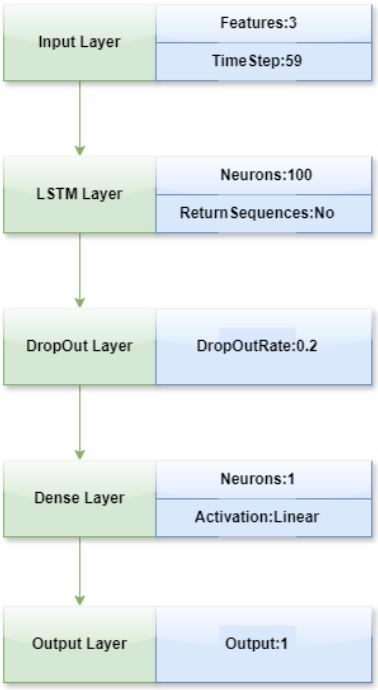

Figure 6. The LSTM structure.

c) Portfolio Optimization

i) Data Processing

We planned to establish a unified rating system for cryptocurrencies, requiring the rating results to reflect the model's predictive performance and the investment value of cryptocurrencies. For the model's predictive performance, since we could not know the actual prices of the day and the future in real investment scenarios, we needed to rely on the model's predicted prices as a reference. Therefore, the predictive performance of the model was particularly important for the rating of a cryptocurrency. For the investment value of cryptocurrencies, in real investment scenarios, due to the strong temporal correlation between cryptocurrency prices and the inevitable deviation between current and future prices, it was necessary to refer to past cryptocurrency prices and other data to determine the current and future investment value of cryptocurrencies. For the scoring system, to reflect the predictive performance of the model and the investment value of cryptocurrency, we needed the following dependent variables as inputs to predict the deviation, true return, return, and volatility. Below are their formulas and handling in special cases:

$$d_t = \hat{y}_t - y_t \tag{11}$$

This is the formula for the deviation value. Among them,

y_t: Represents the true value on the t-th day

$\hat{y}_t$: Represents the predicted value on the t-th day

$$R_t = y_t - y_{t-1} \tag{12}$$

This is the formula for the real profit rate.

$$V_t = \sqrt{\frac{1}{W}\sum_{i=t-W+1}^{t}(x_i - \mu_t)^2} \tag{13}$$

This is the formula for the rolling volatility. For data less than W days, calculate the volatility of all data from the beginning to the current. Among them,

x_i: Represents the i-th data

W: Represents the size of the sliding window

μ_t: Represents the average value within the sliding window on the t-th day. μ_t is calculated as:

$$\mu_t = \frac{1}{W}\sum_{i=t-W+1}^{t} x_i \tag{14}$$

Due to the need to establish a unified scoring system, it was necessary to maintain consistency in the model architecture. Similar to the data processing of previous price prediction models, we divided the data into training and testing sets using α as the allocation ratio, and segment a total of N data points based on window size M to obtain N-M+1 windows. Differently, due to the lack of genuine and objective cryptocurrency ratings, we treated all values of each window as variable x and used pseudo labels as substitutes for variable y. Finally, for all data, standardization was uniformly used for normalization. The formula is as follows:

$$\boldsymbol{x_{norm-t}} = \frac{x_t - \mu}{\sigma} \tag{15}$$

Another thing to note is that in real investment, the actual price of the day cannot be predicted. Thus, for each window, the last value d_t was taken as the mean of all values before the window, V_t was taken as the rolling volatility of the last nine days except for the last day, and the calculation of the true profit margin R_t was based on the predicted value $\hat{y}_t$ of the day instead of the true value y_t of the day.

ii) Model Description

In the optimization of the investment portfolio, to simultaneously evaluate the scores of each currency within a certain period, we adopted a multi task learning model, which enabled us to simultaneously estimate the scores of each currency within a certain period during the optimization process and calculate the corresponding weights through the scores. Its structure is shown in Figure 7.

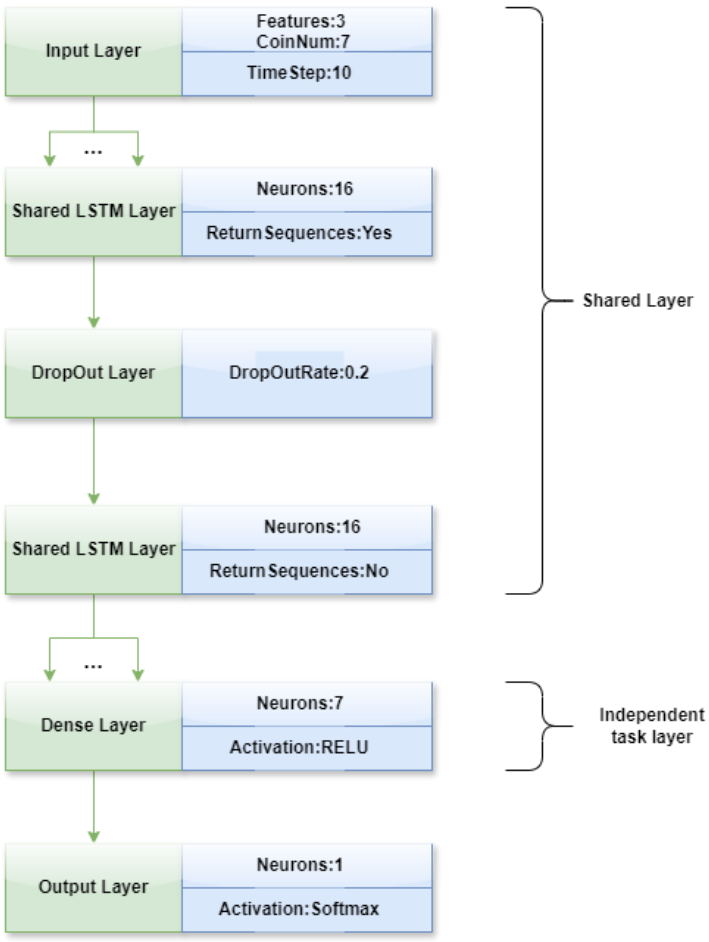

Figure 7. The portfolio optimization model structure.

First, there is the input layer, where Features represents the number of input features (yield, volatility, prediction bias), Coin Num represents the quantity of cryptocurrencies (here are 7 types), and Time Step represents the quantity of input historical data (here predicting the 10th day based on the data from the previous 9 days). This input layer was used to clarify the input format of the data.

Next is the Shared LSTM Layer, where Neurons represent the number of neurons (here, there are 16 neurons, generating 16-dimensional feature values), and Return Sequence represents whether to return a sequence (since subsequent time series processing is required, a sequence needs to be output for subsequent layers to process this data). This layer was used to preliminarily extract time series features, enabling the model to have a rough understanding of the investment value of each currency.

Next is the Dropout Layer, where Dropout Rate represents the random inactivation rate (0.2 in this case, meaning that 20% of neurons in the next layer will no longer work). This layer was designed to make each neuron in the model more independent, enhance robustness, and reduce overfitting.

Next is a Shared LSTM layer, where Neurons represent the number of neurons (here, there are 16 neurons, generating 16-dimensional feature values), and Return Sequence represents whether to return a sequence (which no longer requires temporal processing and is therefore NO). This layer was used to extract temporal features more deeply, giving the model a deeper understanding of each currency's investment value.

Next is the Dense Layer, where Neurons represents the number of neurons (here, there are only 7 neurons, because we needed to output ratings for 7 cryptocurrencies). Activation is the activation function that converts the obtained feature data into the final predicted currency rating.

Finally, there is the Output Layer, where Output represents the number of output features (predicted prices), and Activation is the activation function. This layer uses the activation function and rating to generate the final investment weights.

Four adjustable parameters for parameter selection are time Step, two Neurons, and Dropout Rate. We used a grid search method with a time step of 10, a neuron step of 4, and a Dropout step of 0.05 to gradually find the best parameters.

According to the description in the data processing section, we input the processed data into the same LSTM architecture for the shared model part of each cryptocurrency. This is a neural network model based on multi-task learning. Unlike ordinary neural network models, it introduces a multi-task learning mechanism. On the one hand, it can improve generalization ability and enhance the model's adaptability to new data by sharing features. On the other hand, it can reduce model complexity and overfitting by sharing parameters. Moreover, it can calculate only the weights of the investment portfolio in the same model, enabling us to calculate the Sharpe ratio of the investment portfolio and achieve the loss function mentioned below.

Unlike other conventional LSTMs, our data y here used pseudo labels. Therefore, in this architecture, we adopted a different loss function to train the model in the correct direction instead of randomly bumping around. Furthermore, these cryptocurrencies generally have a higher Sharpe ratio due to the large fluctuations in cryptocurrencies. Therefore, to avoid the weights obtained from the final training being overly biased towards highly volatile cryptocurrencies, we used L2 regularization to sparsify the weights. The L2 regularization formula and loss function are as follows:

$$L_2 = \frac{\lambda}{2}\sum_{i=2}^{n} w_i^2 \tag{16}$$

This is L2 regularization. Among them,

λ : Represents the regularization coefficient (hyperparameter) that controls the strength of regularization

w_i: Represents the i-th weight

$$L(w) = \frac{R_P - R_f}{\sigma_P} + L_2 \tag{17}$$

This is the loss function. Among them,

R_P: Represents the return rate of the investment portfolio. R_P is calculated as

$$R_P = \sum_{i=1}^{n} w_i R_i \tag{18}$$

Among them,

w_i: Represents the i-th weight

R_i: Represents the yield of the i-th cryptocurrency

R_f: Represents the risk-free rate of return. To simplify here, $R_f = 0$

σ_P: Represents the standard deviation (volatility) of portfolio returns. σ_P is calculated as

$$\sigma_P = \sqrt{w^T \sum w} \tag{19}$$

Among them,

w: Represents weight vectors

$\sum$: Represents the covariance matrix of cryptocurrency return rate

For each independent task processing layer of cryptocurrency, we needed to emphasize the importance of price volatility again. Therefore, we directly concatenated their respective volatility with the output results of the previous shared layer and then calculated the score of each currency through the Dense layer. Finally, we input the obtained set of scores into the Softmax layer to calculate weights.

iii) Output Result

After initial processing, each data point is evaluated by an LSTM model (scoring system), and scores are obtained. Then, the score is transformed by the activation function Softmax to form the final investment portfolio. Its formula is as follows:

$$s_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{20}$$

Among them,

s_i: Represents the i-th weight after conversion

z_i: Represents the score of the i-th cryptocurrency before conversion

d) Investment Simulation

In this section, we input the obtained investment weights and their corresponding real profit margins into the calculation formula as follows:

$$M_t = W_t P_t M_{t-1} \tag{21}$$

Among them,

M_t: Represents the total funds owned on the t-th day, and $M_1 = 10000$

W_t: Represents the investment portfolio on the t-th day and $W_t \in R^{1\times n}$

P_t: Represents the real interest rate on the t-th day and $P_t \in R^{n\times 1}$

e) Reliability Analysis

In this section, we established a mathematical model for reliability analysis and planned to use Multi Criteria Decision Analysis (MCDA) to evaluate the risk adjusted return, strategy stability, strategy risk, and strategy effectiveness.

For risk adjusted returns, we measured them using the Sharpe ratio. The formula is as follows:

$$Sharp = \frac{\mu}{\sigma} \tag{22}$$

For strategy stability, we calculated the coefficient of variation by scrolling through the window. The formula is as follows:

$$CV = \frac{\sigma(Sharp - Rolling)}{E(Sharp - Rolling)} \tag{23}$$

For strategic risk, we used $CVaR_\alpha$ to calculate, and the formula is as follows:

$$CVaR_\alpha = -E(R|R \leq VaR_\alpha) \tag{24}$$

Among them,
α: Representative confidence level (taken as 0.95 here)
R: Representative investment return rate
VaR_α: Represents the value at risk at the alpha percentile

To assess the strategy's effectiveness, we observed the distribution of returns for each investment strategy (see Appendix for the investment distribution chart). All investment strategies had the same characteristic: A sharp peak and a thick tail. Therefore, the T-test applicable to normal distribution was no longer used, but the Wilcoxon test was used, with the p-value as the final measure. The Wilcoxon test steps are as follows: First, calculate the paired difference.

$$D_i = Y_i - X_i \tag{25}$$

Remove $D_i = 0$, and leave the remaining sample size n. Second, assign rank R_i to $|D_i|$ to distinguish positive and negative signs. Then, calculate the statistic.

$$W = \min\left(\sum_{D_i>0} R_i, \sum_{D_i<0} R_i\right) \tag{26}$$

Last, calculate the p-value (large sample normal approximation):

$$Z = \frac{W - \frac{n(n+1)}{4}}{\sqrt{\frac{n(n+1)(2n+1)}{24}}} \tag{27}$$

$$p = 2\Phi(-|Z|) \tag{28}$$

Among them,
Φ: Represents the standard normal CDF
Then, the TOPSIS (Approximate Ideal Solution Sorting Method) was used for the final analysis. The TOPSIS calculation steps are as follows: First, standardize the matrix

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} {x_{ij}}^2}} \tag{29}$$

Then, weigh the standardized matrix

$$v_{ij} = \omega_j \cdot r_{ij} \tag{30}$$

Calculate the ideal solution A^+, negative ideal solution A^-

$$A_j^+ = \max v_{ij} \tag{31}$$

$$A_j^- = \min v_{ij} \tag{32}$$

and distance calculation

$$D_i^+ = \sqrt{\sum_{j=1}^{n}\left(v_{ij} - A_j^+\right)^2} \tag{33}$$

$$D_i^- = \sqrt{\sum_{j=1}^{n}\left(v_{ij} - A_j^-\right)^2} \tag{34}$$

Finally, calculate the relative proximity (sorting basis)

$$C_i = \frac{D_i^-}{D_i^+ + D_i^-} \tag{35}$$

The larger the C_i, the better the strategy.
Finally, the entropy weighting method is an objective weighting method based on information entropy, which determines weights by calculating the degree of dispersion (entropy value) of indicators. First, standardize the data.

$$r_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j) + 10^{-10}} \tag{36}$$

Then, calculate the weight.

$$e_j = -\frac{1}{\ln n}\sum_{i=1}^{n}\frac{r_{ij}}{\sum_{i=1}^{n} r_{ij}} \cdot \ln\left(\frac{r_{ij}}{\sum_{i=1}^{n} r_{ij}}\right) \tag{37}$$

Finally, calculate the entropy and weight.

$$\omega_j = \frac{1 - e_j}{\sum_{j=1}^{m}(1 - e_j)} \tag{38}$$

4. Empirical results

4.1. The prediction result of the forecast model

As mentioned, we used closing price, trading volume, and ATR as three dimensions of input data with one day as the time interval. Each currency trained an independent model based on its own dataset, and the target variable for the prediction problem of this model was the daily price of the encrypted currency.

Our training sample size for each dataset was set to 0.7 times the total sample size, while the remaining 0.3 was used as the test sample. Considering that the price scales of different currencies may vary, resulting in different sizes of evaluation indicators used in the end, we did not use MSE, RMSE, or MAE indicators to measure, but instead used R^2 as the indicator.
Its formula is as follows:

$$R^2 = 1 - \frac{\mathrm{SS_{res}}}{\mathrm{SS_{tot}}} \tag{39}$$

Among them,

$\mathrm{SS_{res}}$: The sum of squares of residuals, which represents the sum of squares of the difference between the actual value and the predicted value. It is calculated as:

$$\mathrm{SS_{res}} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{40}$$

$\mathrm{SS_{tot}}$: The total sum of squares, which represents the sum of squares of the difference between the actual value and the mean of the actual value. It is calculated as:

$$\mathrm{SS_{res}} = \sum_{i=1}^{n} (y_i - \bar{y}_i)^2 \tag{41}$$

Among them,

y_i: Represents the actual observation value

$\hat{y}_i$: Represents the predicted value of the model

$\bar{y}_i$: Represents the mean of the actual observed values

n: Represents the number of samples

It can effectively indicate the degree of fit of the model.

During the model tuning process, we found that excessively pursuing a smaller average loss function during training may have led to overfitting of the model. Table 2 shows the fitting of a smaller average loss function.

To solve this problem, we deliberately enlarged the average loss function by reducing the training epochs and other methods. Table 3 shows the fitting situation of the model we chose (please refer to the appendix for the fitting data of a single cryptocurrency).

Table 2. Overfitting model indicator table.

	BCH	BTC	EOS	ETC	ETH	LTC	XRP
Loss	0.0014	4.6e-04	9.3e-04	5.9e-04	3.6e-04	3.8e-04	3.5e-04
R^2	0.955	0.966	0.953	0.864	0.910	0.955	0.911

Table 3. Final model indicator table.

	BCH	BTC	EOS	ETC	ETH	LTC	XRP
Loss	0.0016	5.7e-04	0.0014	0.0012	7.0e-04	8.4e-04	6.7e-04
R^2	0.989	0.980	0.973	0.968	0.970	0.979	0.969

4.2. The optimization results of the investment portfolio model

As mentioned in the section on data processing, the price data predicted by the predictive price model was processed with the actual price data to obtain three dimensions: Deviation, rolling volatility,

and return rate, which were used as input data for the investment portfolio optimization model. Furthermore, the data of these seven cryptocurrencies was divided at a ratio of 0.75, and the training set was input into the investment portfolio optimization model for fitting. The remaining 0.25 test set was then input into the model to obtain daily investment portfolio weights. We displayed the data in the simulation investment phase and compare it with other methods.

4.3. The comparison results between the final investment portfolio and other methods

There are multiple mainstream methods for optimizing investment portfolios in cryptocurrency. They are MV (Mean-Variance Model), MD (Mean-Downside Deviation model), DWP (Dynamic Weighted Portfolio model), CAMP (Capital Asset Pricing Model), and portfolio optimization based on deep learning (referred to as DLS here).

A mainstream investment portfolio optimization method was proposed in a research paper (Markowitz and Todd, 2025). MV is the core model of Modern Portfolio Theory (MPT) proposed by Harry Markowitz. This model constructs the optimal investment portfolio by optimizing the portfolio's expected returns (mean) and risks (variance). For daily cryptocurrency investment portfolios, the calculation process of the MV model is as follows.

First, calculate the expected return. The expected return of an investment portfolio is the weighted average of the expected returns of each asset. Specifically, the expected return of each asset is obtained through historical data or predictive models, and then the overall expected return of the portfolio is calculated based on the weight of the asset in the investment portfolio.

Second, calculate the portfolio risk, which is measured by variance, reflecting the volatility of asset returns. When calculating, it is necessary to consider the weights of each asset and the covariance between assets. Covariance is used to capture the correlation between assets, and diversified investment can reduce non-systematic risk.

Finally, the core of the MV model is used to construct an optimization problem with the goal of minimizing risk given expected returns or maximizing returns given risks. The MD optimization method was proposed in another research (Fabozzi et al., 2011). MD is the mean down bias model and is an improvement of the MV model that focuses on the downside risk of investment portfolios (i.e., the risk of returns falling below a certain target or benchmark). The core idea of the MD model is to construct the optimal investment portfolio by optimizing the portfolio's expected returns and downside risk. Unlike the MV model, which uses variance as a risk measure, the MD model uses downside deviation, which refers to the volatility when returns are lower than the target return. On the one hand, it focuses on downside risk: Investors are more concerned about the risk of returns falling below the target than overall volatility. On the other hand, asymmetric risk measurement: Downward deviation considers only the volatility of negative returns, which is more in line with investors' actual psychology. The calculation process of the MD model is as follows.

First, calculate the expected return of the investment portfolio. The expected return of an investment portfolio is the weighted average of the expected returns of each asset, calculated using the same method as the MV model.

Second, calculate the downward deviation. Downward deviation refers to the volatility of returns below the target return.

Finally, construct an optimization problem. The MD model aims to find a set of weights that minimize downside risk or maximize expected return given downside risk. There are usually two forms: Minimizing downside risk (given expected return) and maximizing return (given downside risk).

DWP is a dynamic weighted portfolio model. It adjusts asset allocation in real-time based on market conditions or asset performance. The core idea of the DWP model is to dynamically adjust asset weights, capture market changes, and improve portfolio returns or reduce risks. Unlike static investment portfolios, DWP models dynamically adjust weights based on predetermined rules or models (such as momentum strategy, mean regression strategy, etc.). This model can dynamically adjust asset weights in real-time based on market conditions or asset performance. It has strong flexibility and can quickly respond to market changes and adapt to different market environments. Diversified investment, by dynamically adjusting weights, further reduces non-systematic risks. The calculation process of the DWP model is as follows.

First, determine the adjustment rules. The core of the DWP model is to dynamically adjust weights, so it is necessary to determine adjustment rules. Common adjustment rules include: Momentum strategy, increasing the weight of assets that have performed well, and decreasing the weight of assets that have performed poorly. Mean regression strategy increases the weight of assets with poor performance and decreases the weight of assets with good performance. Risk parity strategy dynamically adjusts weights based on the risk level of assets, ensuring that each asset contributes equally to the risk of the investment portfolio.

Second, calculate the dynamic weights. Calculate the dynamic weight of each asset according to the adjustment rules. For example, when using a momentum strategy, the momentum indicator of each asset (such as the return rate of the past 12 months) can be calculated, and the weights can be adjusted based on the momentum indicator.

Finally, construct an optimization problem. The goal of the DWP model is to find a set of dynamic weights that maximize the return or minimize the risk of the investment portfolio under given adjustment rules. There are usually two forms: Maximizing returns and minimizing risks.

The CAMP optimization method also optimizes the portfolio value (Sharpe, 1964). CAMP is the capital asset pricing model. It is a model that was proposed by William Sharpe et al. in the 1960s to determine the relationship between the expected returns of assets and their systematic risk (β). The core idea of the CAMP model is to determine the expected return of assets by quantifying their systematic risk (β). It includes systemic risk: The risk of assets is divided into systemic risk and non-systemic risk. Systemic risk is the overall risk of the market and cannot be eliminated through diversified investment; non-systematic risk refers to the risk of specific assets that can be reduced through diversified investments. The relationship between risk and return: The expected return of an asset is directly proportional to its systemic risk (β), with higher β indicating higher expected returns. There is also market equilibrium: In an equilibrium market, the expected returns of all assets are on the Security Market Line (SML). The calculation process of the CAMP model is as follows.

First, the beta value of assets. The beta value is the sensitivity of asset returns to market returns.

Second, calculate the expected return on assets. Finally, construct the Securities Market Line (SML). The Securities Market Line (SML) is a graphical representation of the CAMP model, with β values on the horizontal axis and expected returns on the vertical axis. The slope of SML is the market risk premium. Finally, construct the Securities Market Line (SML). The Securities Market Line (SML) is a graphical representation of the CAMP model, with β values on the horizontal axis and expected returns on the vertical axis. The slope of SML is the market risk premium.

The last popular model we want to compare is the DLS optimization method. As demonstrated (Zhang et al., 2020), a deep learning model is used to directly optimize the Sharpe ratio of the investment portfolio, proposing a framework that avoids the requirement of predicting expected returns and enables direct optimization of portfolio weights by updating model parameters. DLS adopts an LSTM model while adopting a strategy of not predicting prices and uses gradient ascent and new model parameters to optimize weights.

We conducted investment simulations on the data using these five optimization methods. The initial capital was set at \$10000, and leverage was not allowed, short selling was allowed, but the short selling portion of the funds were not used for other investment purposes. To evaluate the performance of our method, we used the following metrics: Standard deviation of return ($Std(R)$), final return rate (Return rate), and maximum deviation of return rate (MDD). As shown in Table 4, the above five methods were compared with our final results. The model we trained here is referred to as the Deep Learning & Multi-task model (**DLM).**

Table 4. The comparison of results from different models.

	$Std(R)$	Return rate(%)	MDD
MD	435.57	−15.9114	−83.5041
CAPM	191.06	187.4306	−46.1786
MV	**127.67**	35.0216	**−43.4826**
DWP	341.13	−5.51152	−79.3075
DLS	327.53	329.9204	−47.8146
DLM	269.26	**507.4994**	−48.9991

A daily comparison of funds for different investment optimization methods is provided, as shown in Figure 8, where Y represents daily held funds, and x represents time (here, the first data point in the dataset of the investment portfolio optimization model is considered the first day).

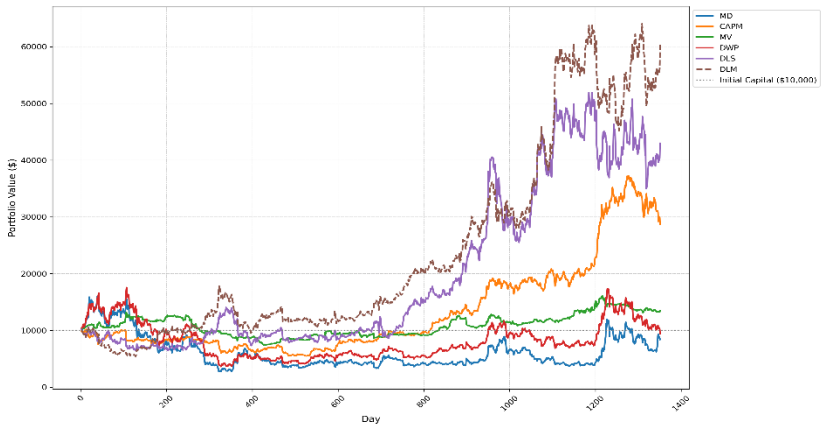

Figure 8. Comparison of different investment strategies.

Then, reliability analysis was conducted, as shown in Table 5, which was the initial indicator for each model.

Table 5. Initial indicators.

	$Sharp$	CV	$CVaR_\alpha$	p
MD	0.018058	4.763584	0.094731	0.842893
CAPM	0.051021	1.640585	0.042701	0.046976
MV	0.021538	**0.24227**	**0.031644**	0.259621
DWP	0.015124	0.634643	0.080779	0.231191
DLS	0.048989	1.952992	0.073523	0.061143
DLM	**0.062865**	1.251308	0.058374	**0.043596**

The final results obtained through Multi Criteria Decision Analysis (MCDA), combined with the Approximate Ideal Solution Sorting Method (TOPSIS) and entropy weight method, are shown in Table 6 and Figure 9.

Table 6. Initial indicators.

	$Weight\_Sharp$	$Weight\_CV$	$Weight\_CVaR_\alpha$	$Weight\_p$	$Score$
MD	0.292294	0.168385	0.214816	0.324504	0.023928
CAPM	0.292294	0.168385	0.214816	0.324504	0.851029
MV	0.292294	0.168385	0.214816	0.324504	0.646256
DWP	0.292294	0.168385	0.214816	0.324504	0.598762
DLS	0.292294	0.168385	0.214816	0.324504	0.782142
DLM	0.292294	0.168385	0.214816	0.324504	**0.879110**

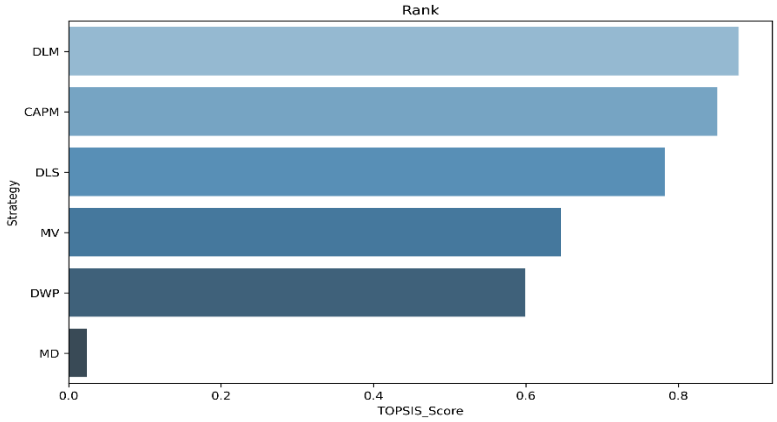

Figure 9. Multiple methods for ranking (the higher the score, the better).

5. Conclusions

In this work, we did not use individual cryptocurrencies but focused on returns and volatility to form an investment portfolio. Considering that price data has a certain relationship over time, we decided to use a neural network LSTM model to optimize the investment portfolio to achieve any effect, such as increasing returns or reducing risk. To enhance this effect, this neural network model adopts a combination of multi-task learning network and LSTM in structure, which can share data of multiple cryptocurrencies and time series relationships, thus unifying the evaluation criteria for cryptocurrencies.

On the loss function, a negative Sharpe ratio was chosen and combined with L2 regularization, enabling the model to balance returns, partial risk, and sparse weights. We selected the past real returns and rolling volatility of cryptocurrencies as input data and added bias values on this basis to reduce the prediction bias of the model. We compared several widely popular algorithms, including MV, MD, DWP, CAMP, and a deep learning-based algorithm called DLS. We tested over a thousand days of data outside the training set. The results indicated that our model typically outperforms other methods in terms of returns, but the variance of returns is also within a controllable range compared to other models. Subsequently, we established a reliability analysis model, which can be said to exhibit the best reliability and efficiency when considering higher risks.

However, we acknowledge that our research has several limitations. First, the generalization ability of our model may be limited by the sample dataset, which comes only from the OKX exchange. Second, our indicator evaluation did not take into account transaction costs, which may have a significant impact on the realized returns in actual transactions. Third, although we used volatility and trading volume as input features, we did not consider other potential influencing factors such as social media sentiment or other macro financial indicators. In future work, we plan to address these shortcomings by exploring more diverse exchange data sources to improve the robustness and applicability of the model. We are also committed to integrating other market features (such as order book microstructure data) and external information (such as news sentiment and macroeconomic indicators) into our modeling framework to improve forecast accuracy.

Author contributions

Weiqi Chen: Conceptualization, Methodology, Software, Validation, Formal analysis, Visualization, Writing- Original Draft and Review. Hang Zheng: Conceptualization, Methodology, Investigation, Resources, Data Curation, Writing- Original Draft and Review, Project administration.

Use of AI tools declaration

The final content and opinions of this article are the authors own responsibilities. Google Translate and ChatGPT are used for correcting and polishing the academic expressions of some contents. Some expressions are polished and better ordered by AI in the Literature Review.

Conflict of interest

The authors declare no conflict of interest.

References

Abu Bakar N, Rosbi S, Uzaki K (2019) Forecasting Cryptocurrency Price Movement Using Moving Average Method: A Case Study of Bitcoin Cash. *Int J Adv Res* 7: 609–614. https://doi.org/10.21474/ijar01/10188

Alahmari SA (2019) Using Machine Learning ARIMA to Predict the Price of Cryptocurrencies. *ISeCure* 11: 139–144. https://doi.org/10.22042/isecure.2019.11.0.18

Angela O, Sun Y (2020) Factors affecting Cryptocurrency Prices: Evidence from Ethereum. IEEE Xplore, 318–323. https://doi.org/10.1109/ICIMTech50083.2020.9211195

Bollerslev T (2008) Glossary to ARCH (GARCH). *SSRN Electron J*. https://doi.org/10.2139/ssrn.1263250

Cohen G (2022) Trading cryptocurrencies using algorithmic average true range systems. *J Forecasting* 42: 212–222. https://doi.org/10.1002/for.2906

Corbet S, Lucey B, Urquhart A, et al. (2019) Cryptocurrencies as a financial asset: A systematic analysis. *Int Rev Financ Anal* 62: 182–199. https://doi.org/10.1016/j.irfa.2018.09.003

Fabozzi FJ, Markowitz HM, Kolm PN, et al. (2011) Portfolio Selection, In: Frank J. Fabozzi, Harry M. Markowitz, *The Theory and Practice of Investment Management: Asset Allocation, Valuation, Portfolio Construction, and Strategies,* Second Edition, 45–78. https://doi.org/10.1002/9781118267028.ch3

Ferland R, Latour A, Oraichi D (2006) Integer-Valued GARCH Process. *J Time Series Anal* 27: 923–942. https://doi.org/10.1111/j.1467-9892.2006.00496.x

García-Medina A, Aguayo-Moreno E (2024) LSTM–GARCH Hybrid Model for the Prediction of Volatility in Cryptocurrency Portfolios. *Computat Econ* 63: 1511–1542. https://doi.org/10.1007/s10614-023-10373-8

García-Medina A, Luu Duc Huynh T (2021) What Drives Bitcoin? An Approach from Continuous Local Transfer Entropy and Deep Learning Classification Models. *Entropy* 23: 1582. https://doi.org/10.3390/e23121582

Golnari A, Komeili MH, Azizi Z (2024) Probabilistic deep learning and transfer learning for robust cryptocurrency price prediction. *Expert Syst Appl* 255: 124404. https://doi.org/10.1016/j.eswa.2024.124404

Ji S, Kim J, Im H (2019) A Comparative Study of Bitcoin Price Prediction Using Deep Learning. *Mathematics* 7: 898. https://doi.org/10.3390/math7100898

Khedr AM, Arif I, El‐Bannany M, et al. (2021) Cryptocurrency price prediction using traditional statistical and machine-learning techniques: A survey. *Intell Syst Account Financ Manage* 28: 3–34. https://doi.org/10.1002/isaf.1488

Kwon DH, Kim JB, Heo JS, et al. (2019) Time Series Classification of Cryptocurrency Price Trend Based on a Recurrent LSTM Neural Network. *J Inf Process Syst* 15: 694–706. https://doi.org/10.3745/JIPS.03.0120

Liu W (2018) Portfolio diversification across cryptocurrencies. *Financ Res Lett* 29: 200–205. https://doi.org/10.1016/j.frl.2018.07.010

Ma Y, Ahmad F, Liu M, et al. (2020) Portfolio optimization in the era of digital financialization using cryptocurrencies. *Technol Forecast Soc* 161: 120265. https://doi.org/10.1016/j.techfore.2020.120265

Malsa N, Vyas V, Gautam J (2021) RMSE calculation of LSTM models for predicting prices of different cryptocurrencies. *Int J Syst Assur Eng Manage*, 1–9. https://doi.org/10.1007/s13198-021-01431-1

Markowitz H, Todd GP (2025) *Mean-Variance Analysis in Portfolio Choice and Capital Markets*, John Wiley & Sons. Available from: https://books.google.com.hk/books?hl=zh-CN&lr=&id=eJ8QUsgfZ8wC&oi=fnd&pg=PR9&dq=mean-variance&ots=t7tXVk4iex&sig=B7byDsUcFA1zhGjcrtquA6y2asU&redir_esc=y#v=onepage&q=mean-variance&f=false.

Patel MM, Tanwar S, Gupta R, et al. (2020) A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions. *J Inf Secur Appl* 55: 102583. https://doi.org/10.1016/j.jisa.2020.102583

Petukhina A, Trimborn S, Härdle WK, et al. (2021) Investing with cryptocurrencies – evaluating their potential for portfolio allocation strategies. *Quant Financ* 21: 1–29. https://doi.org/10.1080/14697688.2021.1880023

Ruder S (2017) An Overview of Multi-Task Learning in Deep Neural Networks. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.1706.05098

Sharpe WF (1964) Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk. *J Financ* 19: 425–442. https://doi.org/10.1111/j.1540-6261.1964.tb02865.x

Wen NS, Ling LS (2023) Evaluation of Cryptocurrency Price Prediction Using LSTM and CNNs Models. *JOIV Int J Inf Visual* 7: 2016–2016. https://doi.org/10.30630/joiv.7.3-2.2344

Wu CH, Lu CC, Ma YF, et al. (2018) A New Forecasting Framework for Bitcoin Price with LSTM. IEEE Xplore. https://doi.org/10.1109/ICDMW.2018.00032

Zhang Z, Zohren S, Roberts S (2020) Deep Reinforcement Learning for Trading. *J Financ Data Sci* 2: 25–40. https://doi.org/10.3905/jfds.2020.1.030

Zhao D, Rinaldo A, Brookins C (2019) Cryptocurrency Price Prediction and Trading Strategies Using Support Vector Machines. ArXiv.org. https://doi.org/10.48550/arXiv.1911.11819

AIMS Press