Expert Systems With Applications 237 (2024) 121520

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Attention-based CNN–LSTM for high-frequency multiple cryptocurrency trend prediction

Peng Peng [a,d], Yuehong Chen [b,*], Weiwei Lin [c,d,**], James Z. Wang [e]

[a] School of Future Technology, South China University of Technology, Guangzhou, 510000, China
[b] School of Mathematics and Systems Science, Guangdong Polytechnic Normal University, Guangzhou, 510665, China
[c] School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510000, China
[d] Peng Cheng Laboratory, Shenzhen, 518000, China
[e] School of Computing, Clemson University, SC, USA

ARTICLE INFO

Keywords:
Time-series forecasting
Cryptocurrency
Labeling method
Attention mechanism

ABSTRACT

With the price of Bitcoin, Ethereum, and many other cryptocurrencies climbing, the cryptocurrency market has become the most popular investment area in recent years. Unlike other relatively more stable financial derivatives, the cryptocurrency market has high volatility which requires a high-frequency prediction model for quantitative trading. However, the excessive number of trading becomes a critical issue due to the instability of the prediction results and high error rate. To relieve such a problem, based on the observation of high-frequency data, we use local minimum series to replace the original series and propose a more stable triple trend labeling method that reduces the number of trades by potentially influencing the training of the model. In addition, a new attention-based CNN–LSTM model for multiple cryptocurrencies (ACLMC) is proposed to optimize model effects by exploiting correlations across frequencies and currencies, and to smooth out the investment risk associated with prediction errors by supporting simultaneous multi-currency predictions. Experiments show that our labeling method with ACLMC can achieve much better financial metrics and fewer number of transactions than traditional baselines.

1. Introduction

Nakamoto (2008) proposed the first cryptocurrency, the Bitcoin (BTC). The rapid climbing price of Bitcoin from US$0.08 in 2010 to the highest point of US$69 000 on 2021-11-10, attracts an extremely large number of investors and developers. Unlike traditional investment areas, the cryptocurrency market has more volatility and is more susceptible to perturbation by various factors. Meanwhile, cryptocurrencies are traded 24/7, so finding a stable quantitative trading strategy can effectively help investors gain income.

Machine learning methods have shown the powerful ability of time series prediction and have been successfully adopted in gold (Vidal & Kristjanpoller, 2020), stock (Selvin et al., 2017), and forex price prediction (Yıldırım et al., 2021). Generally speaking, these financial derivatives are more stable, meaning that their prices are less likely to fluctuate significantly in a short period of time. Low-frequency trading strategies can already generate attractive returns. However, cryptocurrencies are often subject to large price changes. For example, since May 5, 2022, the price of the LUNA coin has fallen by more than 99% in a short period of time and has led to a rapid decline in the price of other cryptocurrencies, such as the price of BTC, which fell from approximately 37,000$ to about 20,000$ on June 17. Therefore, a high-frequency price prediction method and trading strategy are required in the cryptocurrency market.

Some existing work has considered this issue and proposed various methods to perform high-frequency forecasting, such as predicting prices or identifying uptrends or downtrends in the next few minutes (Zhang et al., 2021). However, only a few works give the number of transactions in simulating investment returns, which is a serious impact to consider in a high-frequency environment. Typically trading strategies combined with forecasting methods buy or sell positions as the forecast changes, so forecast series stability and model accuracy are key to reducing the number of trades. However, current methods are generally difficult to obtain stable forecast series, which we believe is due to the stochastic nature of financial series, as shown in Fig. 1. When in a sideways oscillation range, small random up-and-down price fluctuations occur, which are more common in the high-frequency case. Traditional labeling methods define the forecast trend target as the price difference between two points, which can confuse the model,

* Corresponding author at: School of Mathematics and Systems Science, Guangdong Polytechnic Normal University, Guangzhou, 510665, China.
** Corresponding author at: School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510000, China.
E-mail addresses: pengp@pcl.ac.cn (P. Peng), yhchen1997@126.com (Y. Chen), linww@scut.edu.cn (W. Lin), jzwang@clemson.edu (J.Z. Wang).

https://doi.org/10.1016/j.eswa.2023.121520
Received 6 August 2022; Received in revised form 20 June 2023; Accepted 7 September 2023
Available online 16 September 2023
0957-4174/

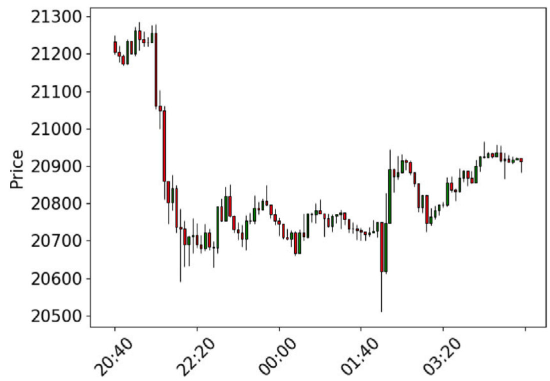


Fig. 1. BTC 5 min k-line graph from 2022-06-27 20:40:00 to 2022-06-28 05:00:00. The price shows stochasticity between 22:20 to 01:40.

making it less accurate and leading to an unstable sequence of forecast results.

Therefore, we first propose a novel triple classification trend labeling method. Inspired by Zhao et al. (2018), the local minimum series can represent the original price series well while eliminating the randomness. The gap between the true series and the local minimum series is minimized in the high-frequency setting. Moreover, directly reducing the number of possible transaction points is also an intuitive way to reduce the number of trades. After that, we divide trends into three classes, increase, decrease, and stable, where the stable trend means the price is under a minor fluctuation interval and the strategy should avoid trading. We further optimize the label definition to allow fluctuations under a certain threshold within the trend interval. Such an approach provides a more stable sequence of labels, alleviates stochasticity, helps to train the model, and potentially enables the model to output a low volatility prediction sequence, reducing the number of transactions.

Further, a novel Attention-based CNN–LSTM model for Multiple Cryptocurrencies(ACLMC) is proposed, which combines convolutional neural network(CNN) and long-short term memory network(LSTM) that has been proven to have a better capability through extensive practice, and adopt attention modules to explore more correlation of various currency and frequency data. Firstly, we design attention-based CNN–LSTM encoders to extract hidden features for each currency and frequency of data, while encoders for the same frequency data are weight-sharing to learn general information about different currencies. Second, to combine information from multiple frequencies and currencies, we include two attention layers to explore and selectively integrate features. In addition, multiple predictors are used to simultaneously make predictions, which facilitates model training and eventually supports simultaneous trading of multiple currencies, thus further optimizing financial indicators and reducing investment risk.

The main contributions are summarized as follows:

- Local minimum series is used and a much more stable triplet trend labeling methodology is proposed to optimize the unacceptable number of transactions under high-frequency settings.
- We propose the ACLMC model, which combines data from different frequencies and different cryptocurrencies through the attention module to improve the accuracy of predictions. Simultaneous output of multiple currencies can support simultaneous trading of multiple currencies and smooth out the investment risks associated with errors in individual currency predictions.
- In our experiments, we demonstrate that the strategy based on the proposed trend labeling approach with the ACLMC model is much better than the traditional ones in multiple investment indicators and prove the effectiveness of both components separately.

The rest of this paper is structured as follows. In Section 2, we introduce and discuss some related works about labeling methods and financial price series prediction. Section 3 describes our labeling method with the ACLMC model. Experiments about our methods can be found in Section 4, which compares our method with other famous baseline methods and prove the advantages of what we propose. Section 5 gives a conclusion of this paper and points out some future directions of research.

2. Related work

2.1. Trend labeling method

Trend forecasting can be divided into two main directions, regression, and classification. Regression implies modeling the price change and predicting the actual future price or the price difference between current and future. Normally, the next time closing price is directly used as the current predicted label (Dutta et al., 2020). However, for investors, gaining returns is more about the relative prices of two time points than the true price of a particular point. Thus, the regression-based approach does not match what investors expect, and more researchers turn to the classification direction.

The target of classification is to predict the future trend based on the difference between two points. The most widely used is to mark as up when the next interval price is greater than the current, and fewer are marked as down (Livieris et al., 2021). After this, some works further define a stable label which means the absolute value of the difference of prices between two intervals is under a given threshold. For example, Wu et al. (2021) simply defines a percentage price change of no more than 0.05 as no increase or decrease, and Shintate and Pichl (2019) using the logarithmic return between the current price and the future price, and dividing it equally into three categories, up, down, and static. Yet, these methods cannot deal with the stochastic financial time series that seriously affects the stability of the label, which in turn makes the model effect decline. Therefore, Wu et al. (2020) proposed a binary trend labeling method, treating peaks and troughs that do not exceed a certain fluctuation threshold as continuous trends. Experiments show that compared with traditional classification labeling methods, their trend labeling can enhance the net yield to maturity.

2.2. Price series prediction

The development of the machine learning algorithms such as logistic regression, support vector machine, and decision tree bring the possibility of modeling the price series (Hu et al., 2021). In recent years, most of the work use the recurrent neural network and the long-short term memory network as the backbone to extract time series hidden features in the stock price prediction field. For example, Ding and Qin (2020) considers the relationship between the opening price, the lowest price, and the highest price and builds the associative network based on LSTM for simultaneous prediction. Liang et al. (2019) apply the multioptical combination wavelet transform for denoising and predicting stock price through LSTM. Li et al. (2018) modifies the LSTM structure to support multiple inputs through the attention mechanism, thus enhancing the model performance with data from the relevant stocks. Natural Language Processing(NLP) also provides a new direction to enhance prediction accuracy. Some works utilize the sentiment analysis on social media or news to discover the traders' investment intentions and the relationship to the financial derivatives (Matsubara et al., 2018; Mohan et al., 2019; Zhao et al., 2020).

Some works also turn to the Convolutional Neural Network(CNN) for better local feature extracting ability. For each day, Sezer and Ozbayoglu (2018) concatenates 15 distinct indicators with 15 different time intervals to form a 15×15 image and uses CNN for prediction. To better utilize the temporal information, the CNN–LSTM network is built with a CNN module to extract daily features and an LSTM module to process the time relationship of series features. Rezaei et al. (2021) uses frequency decomposition for the price series, extracts the features by CNN–LSTM separately, and aggregates the prediction results. Lu et al. (2021) employs CNN-BiLSTM for feature extraction and uses an attention mechanism to fuse the output of different time points of BiLSTM to enhance the effect of temporal information extraction. Similarly, Livieris et al. (2021) exploits the feature extraction capability of CNN–LSTM to extract information about different cryptocurrencies and predict the price of a certain currency after concatenation. All of them achieve better accuracy than only using CNN or LSTM.

To further enhance the performance of prediction, noise-related analysis attracts many concerns, which tries to relieve and eliminate noise or outliers influence in the original price series. Ma et al. (2022) propose to apply the denoising autoencoder to generate correct labels without noise exposure, which is trained via minimizing the reconstruction loss against the pure input organized by simple moving averages(SMA) and exponential moving averages(EMA). However, since the SMA and EMA series are generated by the noisy price series, the module performance cannot be ensured. Ozer and Okan Sakar (2022) represents the distance between two series by dynamic time warping technique and uses k-means clustering to locate cluster centers. Outliers far from the centers are removed from the training set to reduce noise and experiment results show the great performance of their methods combined with various machine learning approaches. However, when a new market style began to form, their k-means clustering module may fail to recognize it in time.

3. Methodologies

3.1. Price trend labeling method

Traditional labeling methods are challenging to be applied directly to high-frequency cryptocurrency forecasting problems. High-frequency trading setups provide more trading opportunities compared to low-frequency trading setups, ideally avoiding losses and increasing profits, but too many trading opportunities may also result in an unacceptable number of transactions and transaction fee overhead. Most of the existing models are unable to achieve satisfactory prediction accuracy, and it is difficult to significantly improve the model accuracy and thus increase the revenue to cover the transactions overhead. Therefore, how to improve the stability of the model prediction series and how to appropriately combine high-frequency trading time points become the optimization direction to alleviate this problem.

First, it is suggested to replace the original series with a local minimum series to mitigate the stochasticity of the financial series and to merge the high-frequency trading time points. Specifically, for the closing price series $P = \{p_0, p_1, \dots, p_t\}$, we can select a series of local minimum points that satisfies $p_{\hat{i}} < p_{\hat{i}-1}$ and $p_{\hat{i}} < p_{\hat{i}+1}$ and form a local minimum series $\hat{P} = \{p_{\hat{0}}, p_{\hat{1}}, \dots, p_{\hat{t}}\}$.

Theoretically, there may be a large difference between the original closing price series and the local minimum series, causing a loss of valid information while cutting the effect of stochasticity, especially when there is a continuous rise and fall, the local minimum series has a risk of losing information on peaks and troughs. Fortunately, in a high-frequency environment, higher stochasticity can effectively break the continuous variation and create more local minimum points, making the gap between these two series significantly narrower to an acceptable range. Fig. 2 gives an example using a randomly selected 5-min frequency data segment of Binance Coin (BNB) with the original closing price series plotted in red and the local minimum series plotted in blue for comparison.

After that, we propose a trend labeling method to further remove the randomness and increase the stability of the labeling series. The method adjusts the prediction target of the model from the original short-term up-and-down prediction to the long-term trend prediction,

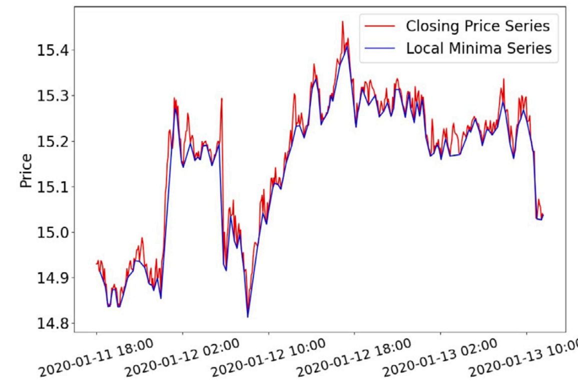


Fig. 2. BNB 5 min k-line graph. The red series represent the original close price and the blue series is the local minimum series.

which can avoid the unstable labels under random perturbation to disturb the model and improve the stability of the model prediction result series. We use a triple-category label, adding a stable category instead of the traditional binary category, which can reduce transaction overhead by avoiding unnecessary large numbers of trades during periods of sideways market oscillations.

Algorithm 1 demonstrates our proposed auto-labeling algorithm. In detail, firstly, the local minimum series is obtained from the original time series. At the beginning of the automatic labeling method, the two hyperparameters θ_c and θ_s are defined as thresholds for the trend change and the stability range respectively. We use d to denote the current trend, where the upward trend is 1, the downward trend is −1, and the stable trend is 0, and set the default value at the beginning of the algorithm to $d = 1$. The algorithm manages an interval with an index range from s to e, where e is the index of extreme points, i.e. the highest price point in the current interval when $d = 1$, and the lowest when $d = -1$.

The algorithm traverses the entire data sequence and performs the following actions. If the current price is not in the local minimum series, simply continue processing. Otherwise, it checks whether the current price breaks the trend of the current interval. If $(p_i - p_e)/p_e * d > -\theta_c$, we consider it as not breaking the trend, then update extreme point information if needed and continue processing. If it breaks the trend, go back to the last extreme point e, allocate labels for the interval between s to e, reverse the trend d, reset s and e to e, and continue the iteration from e. For label allocation, if the fluctuation of the interval is small than the threshold, i.e. $|(p_e - p_s)/p_s| \leq \theta_s$, assign it to label 0 for the stable interval, otherwise assign label d for the upward or downward interval.

3.2. ACLMC model

Before introducing our ACLMC model, it is necessary to give a brief review of CNN, LSTM, and the attention module. CNN is a widely used module firstly proposed in 1998 for feature extraction (LeCun et al., 1998). Nowadays, CNN has shown the powerful ability in computer vision(CV), natural language processing(NLP), and the time series processing fields (Alzubaidi et al., 2021). Normally, CNN contains the convolution layer, the pooling layer, and the full connection layer. In the convolution layer, multiple convolution kernels are trained to percept and extract local features by steply scanning the input matrix, as in (1), where x is the input vector or matrix, w_k, b_k are parameters of the kth kernel, f is the activation function, and h_k is the hidden feature extracted by the kth kernel. The pooling layer is used to down-sample the features from the convolution layer, thus reducing computational costs and avoiding overfitting. The full connection layer flattens the features to a vector and linearly maps it to get the target.

$$h_k = f(w_k * x + b_k) \tag{1}$$

Algorithm 1 Auto-Labeling Algorithm

Input: $\theta_c, \theta_s, P = \{p_0, p_1, ..., p_t\}$
Output: Labels of P

```
// Obtain local minimum series
Initialize local minimum series {p_0}
for p_i in P do
    if p_{i-1} > p_i and p_i < p_{i+1} then
        Append p_i to local minimum series
    end if
end for
Initialize s ← 0, e ← 0, p_e ← p_0, d ← 1
Initialize search pointer i ← 0
while i < t do
    if p_i not in local minimum series then
        Continue
    end if
    if (p_i − p_e)/p_e ∗ d > −θ_c then
        // If NOT break the trend
        Update extreme point information
        Continue
    else
        if |(p_e − p_s)/p_s| ≤ θ_s then
            // If interval fluctuation smaller than threshold
            Update labels of current interval from s to e as 0
        else
            Update labels of current interval from s to e as d
        end if
        Update d ← d × −1, s ← e, ptr ← e
    end if
    i = i + 1
end while
return Labels of P
```

The Recurrent Neural Network(RNN) is widely used in time series problems because of the ability to handle temporal information, but the long-term dependency problem, the gradient explosion, and the gradient vanishing phenomenon severely limit the model performance. To solve these, Hochreiter and Schmidhuber (1997) proposes LSTM, which additionally transmits cell state and hidden state within the time series. The LSTM also controls the ratio of past memories to new data by three gating signals (forgetting gate, input gate, and output gate). For inputs x_t in time t, LSTM concatenate it with the previous hidden state output h_{t-1} of the time $t-1$ to calculate the three gating signals and one input state as (2), where $Z_{forget}, Z_{input}, Z_{output}$ represent the forgetting gate, input gate, output gate respectively, i_t is the input state, σ is the sigmoid function, and $tanh$ is the hyperbolic tangent function.

$$
\begin{aligned}
Z_{forget} &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\
Z_{input} &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\
Z_{output} &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\
i_t &= tanh(w[h_{t-1}, x_t] + b)
\end{aligned}
\tag{2}
$$

LSTM can be divided into the following three steps. Firstly, the forgetting gate controls the percentage of cell states that will be preserved. Then selectively record and update x_t and h_{t-1} into the cell state by the input gate, as shown in (3), where the $\circ$ represents the element-wise product. Finally, The output gate controls which part of the updated cell state c_t is used as the output hidden state h_t. Also, h_t can be processed to obtain the final output y_t of the network, as shown in (4).

$$c_t = c_{t-1} \circ Z_{forget} + i_t \circ Z_{input} \tag{3}$$

$$
\begin{aligned}
h_t &= tanh(c_t) \circ Z_{output} \\
y_t &= f(h_t)
\end{aligned}
\tag{4}
$$

The attention module is a famous module nowadays to explore the relationship between the input feature vectors, which can be described as mapping a query and a set of key–value pairs to an output (Vaswani et al., 2017). For the query feature vector q_i and the feature vectors $h_1, \dots, h_j$ to be merged, each q_i is linearly projected to the query q_i, while each h_i is linearly projected to the key k_j, and the value v_j. For query q_i, we measure the correlation between q_i to each key k_j and use the softmax function to get the correlation weight between them. The final attention result r_i is the weighted sum of the values plus the original query vector q_i. (5) gives the formal description.

$$
\begin{aligned}
w_{ij} &= \frac{\exp(k_j^{\top} q_i)}{\sum_{j=1}^{n} \exp(k_j^{\top} q_i)} \\
r_i &= q_i + \sum_{t=1}^{n} w_{ij} v_j
\end{aligned}
\tag{5}
$$

Our ACLMC model structure is shown in Fig. 3. Generally, a weight-sharing encoder is adopted to extract hidden features from the original data. After that, a frequencies attention layer is used to combine the hidden feature of different frequency data. The currencies attention layer is utilized to discover useful information between data of other currencies. To get the final prediction result, multiple predictors are designed for different coins. The overall structure of the model is shown in Fig. 3 and the details will be described afterward.

For financial data, a large number of trading data are gathered by a given frequency and transformed into various financial indicators. For currency i at time t, based on a given frequency, such as 5-min, by collecting all trading data from $t-5$ min to t, we can calculate a series of financial indicators, and group them into a vector, denoting as $\boldsymbol{x}^t_{i,5\ \min}$. Similarly, 1-day frequency data can be obtained and denoted as $\boldsymbol{x}^t_{i,daily}$. In our approach, data with two frequencies of c currencies are considered for performance improvement. Meanwhile, we also want to explore and utilize temporal information. Therefore, the input of our model of time t can be denoted as $\boldsymbol{X} = \{\boldsymbol{x}^{t-\delta}_{i,daily}, \boldsymbol{x}^{t-\delta}_{i,5\ \min} | \forall i \in 1, \dots, c, \forall \delta \in 0, \dots, \Delta\}$, where Δ is the look-back length.

Firstly, we utilize two weight-sharing CNN–LSTM modules to separately extract features of data with different frequencies. The weights are shared because we expect the encoder to learn information about different currencies and optimally compress the high-dimensional raw features consisting of multiple financial factors into features that are more relevant to the prediction target. Considering the great local feature extraction ability of CNN and the temporal information processing ability of LSTM, we combine these two modules to achieve better effectiveness. The convolution and pooling layers are processed along the feature vector, focusing on local features within the vector, after which the LSTM attempts to gradually fuse the historical data as it moves forward in time. Eventually, inspired by Wang et al. (2016), a composite feature is finally obtained by mining the temporal correlation backward through the attention mechanism. The models are denoted as $E_{5\ \min}$ and E_{daily}, with different parameters $\Theta_{5\ \min}$ and Θ_{daily} but the same structure in the frequency dimension, and shared parameters in the currency dimension. Therefore, the extracted hidden feature of currency i at time t can be computed by (6).

$$
\begin{aligned}
h^t_{(i,5\ min)} &= E_{5\ \min}(\{\boldsymbol{x}^{t-\delta}_{(i,5\ min)} | \forall \delta \in 0, \dots, \Delta\} | \Theta_{5\ \min}) \\
h^t_{(i,daily)} &= E_{daily}(\{\boldsymbol{x}^{t-\delta}_{(i,daily)} | \forall \delta \in 0, \dots, \Delta\} | \Theta_{daily})
\end{aligned}
\tag{6}
$$

To further explore the correlation between different statistical frequency data and currencies, the traditional method is to simply concatenate these feature vectors and implicitly handled them in the follow-up network. However, this method constantly extracts features from other data and is unable to selectively fuse features based on the current data. For example, at different points in time, the relationship between BTC hidden features and Ethereum(ETH) hidden features is most likely not fixed, but changes in real-time with the correlation of the two features. Therefore, we use the attention mechanism to achieve this requirement. Formally, the hidden features of different

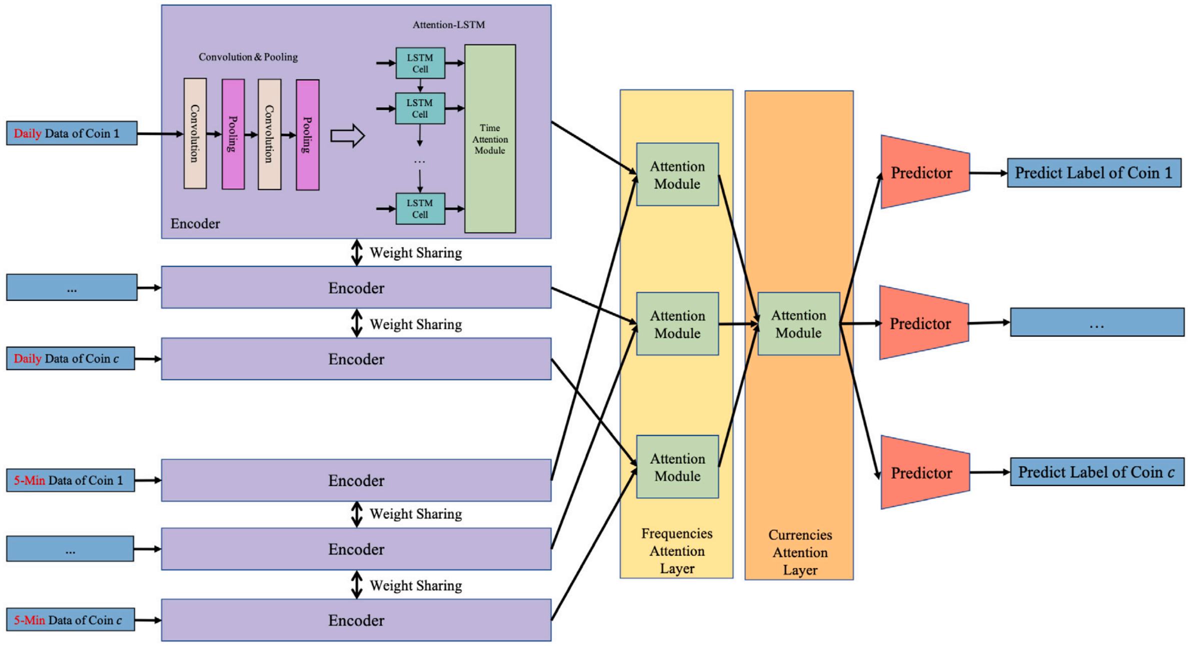


Fig. 3. Attention-based CNN–LSTM model for multiple cryptocurrencies (ACLMC) structure. The weight-sharing encoder extracts feature from different frequency data separately. The frequency attention layer combines features from different frequencies but the same currency, and the currency attention layer further explores cross-currency information. Multiple predictors for each currency are designed to output simultaneously.

statistical frequency are merged firstly by the attention module with parameters σ_f. After that, for each currency corresponding feature vector, the attention operation with parameter σ_c is performed with all other currencies to get the final result, as shown in (7), where the $Attn(\boldsymbol{x}, \boldsymbol{Y}|\sigma)$ represent the attention operation from vector $\boldsymbol{x}$ to each vector in $\boldsymbol{Y}$ with parameter σ.

$$h_i^t = Attn(h_{(i,5\ min)}^t, h_{(i,daily)}^t|\sigma_f)$$
$$h^t = Attn(h_i^t, \{h_j^t|\forall j \in 1, \dots, c\}|\sigma_c) \tag{7}$$

Finally, for each currency, we use a multilayer perceptron P_i including multiple linear hidden layers to obtain predictions and use a Softmax function to transform the output vector into a triple classification probability. Same structure but different parameters is used for different currencies to get the final prediction results, as shown in (8), where ϕ_i denotes the parameters of currency i. The cumulative cross-entropy loss of all currencies with back-propagation is used to optimize the model parameters.

$$\hat{y}_i = P_i(h^t|\phi_i) \tag{8}$$

4. Experiments

4.1. Datasets

Thanks to the blockchain-based nature of cryptocurrencies, trading data can be found on various trading sites. All data used in experiments are collected from the Binance Data. We selected the top five currencies sorted by market value to obtain multiple inputs. Except for TetherUS(USDT) and USD Coin(USDC) which is 1 : 1 pegged to the U.S. dollar, Bitcoin(BTC), Ethereum(ETH), Binance Coin(BNB), Ripple(XRP), and Cardano(ADA) are finally chosen. The dataset contains trade data from 2020-01-01 to 2022-06-01. The reason we chose this time period to comprise our data set is that starting in 2020, cryptocurrencies begin to attract investor attention on a large scale and prices begin to see significant increases. Series of financial indicators are made based on the trade data in 5-min or daily statistical intervals respectively, listed in Table 1. Because of the large gap between currencies, we utilize the min–max normalization for each currency for data standardization. The initial training set contains data from 2020-01-01 to 2021-11-01, and the test set ranges from 2021-11-10 to 2022-06-01.

Table 1
Applied financial factors.

Basic indicators	Opening price Closing price High price Low price
Price related indicators	Moving average Relative strength index Bollinger band Moving average convergence divergence Average directional movement index Commodity channel index Rate of change
Volume related indicators	Chaikin A/D line On balance volume Parabolic SAR Money flow index Volume Active buy turnover Net active buy turnover Active buy turnover ratio Net active buy turnover ratio Active buying strength Net active buying strength The proportion of big active buy order The proportion of big active sell order Delta proportion of big buy/sell order The proportion of big order The concentration ratio of buy order The concentration ratio of sell order Order concentration difference Sum of buy and sell order concentration High frequency skewness Volume-price correlation Value of big order that push the price Percentage of downside volatility

4.2. Experimental settings

The hyper-parameters, the experiment platform, and implementation details are presented as follows. All the experiments are based on a CentOS server with NVIDIA 1080Ti GPU. Pytorch with version 1.11.0 is

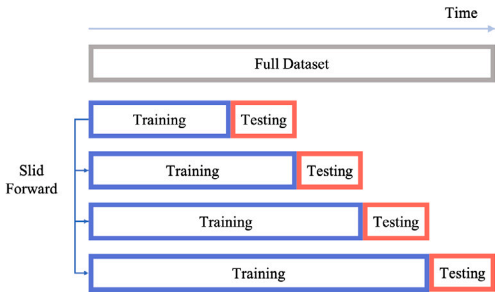


Fig. 4. Schematic of sliding window testing.

used. For the labeling method, in order to balance the proportion of the number of samples with different labels, θ_c, θ_s are set to 0.03 and 0.06 respectively. For the hyper-parameters of ACLMC, after fine-tuning, the learning rate is set to 0.001 with cosine scheduler (Loshchilov & Hutter, 2016) and the optimizer is Adam (Kingma & Ba, 2014) with default settings of Pytorch. The batch size is 128 and the number of epochs is 20 with early stopping. The input vector series length is set to 30 with a total of 42 factors. The hidden layer dimension is set to 128. Similar to most existing works (Li & Wu, 2022; Zhang et al., 2021), the sliding window is used for testing, as shown in Fig. 4. For each step, only the test data in the sliding window is predicted, and then this part of the data is added to the training set and slid the window forward. After all the results of the test set are obtained, simulated trades are performed through the results of the entire test set, and evaluation metrics are calculated. Thus, the result values computed by this method actually aggregately reflect the performance of the model with different training and test sets. In this work, the length of the sliding window was set to 2880 five-minute time points, i.e. 10 days.

4.3. Evaluation metrics

Most of the existing work evaluates the comparative effects from two perspectives, classification metrics, and financial metrics. However, we find that the correlation between classification metrics and financial metrics is weak during training. We believe this is because financial indicators are not only influenced by the accurate number of classifications, but also by the accurate position. A correct forecast in a position with high volatility brings significantly different returns than a correct forecast in a position with low volatility. Therefore, models that tend to correctly judge large volatility positions may have similar classification indicators and significantly different financial indicators than models that tend to correctly judge small volatility positions.

We adopt two commonly used trading strategies, buy-hold, and long-short for simulation.We assume the initial cash is 1, and totally n currencies can be traded. The buy-hold strategy is the simplest and most intuitive way to trade currencies. Specifically, it spends current cash to buy currencies when the algorithm predicts prices will rise, sells all holding currencies when prices are predicted to fall, and not trading when prices are predicted to remain stable. In addition, the cryptocurrency market supports a long-short trading strategy. The difference between long-short strategy and buy-hold strategy is that long-short strategy not only allows to buy currencies at low prices and sell them at high prices, i.e. long positions, but also allows to borrow and sell currencies at high prices and buy them back at low prices, i.e. short positions. Thus, the algorithm returns all borrowed currency when the price is predicted to rise and spends current cash to buy currency, or sells the holding currencies when the price is predicted to fall and borrows currency to sell. Note that in order to accommodate the above-mentioned replacement of the original price series by local minimum for label formulation, model training, and forecasting, here we also simulate trading at local minimum points only. Since the local minimum actually needs to be judged at $t + 1$ at time t, we trade the corresponding predicted results for t only at $t + 1$ to avoid leakage of future data.

The following financial metrics are selected. First, test period excess return(ER) is the most straightforward metric to evaluate. This indicates the return above the benchmark strategy and the larger the better. Second, under the high-frequency setting, the number of transactions(NT) is also an important part. A smaller number of transactions means smaller transaction overhead. The Sharpe Ratio(SR), also known as the Sharpe Index, which is designed to calculate the return over the risk-free return for every unit increase in risk, is used for comparison. A higher SR means that a strategy can achieve a higher return with lower risk and can effectively evaluate the performance of a strategy. (9) gives the formula of Sharpe Ratio, where R_s, R_{rf}, σ_s represent the strategy return, risk-free profit, and the standard derivation of strategy return respectively. In addition to this, maximum drawdown (MDD) is a measure of an investment's largest price drop from a peak to a trough that shows the relative riskiness of strategies. (10) gives the formula of MDD, where the Trough Value is the highest value the strategy obtained when trading and the Peak Value is the smallest one.

$$SharpeRatio = \frac{E(R_s) - R_{rf}}{\sigma_s} \tag{9}$$

$$MDD = \frac{\text{Trough Value} - \text{Peak Value}}{\text{Peak Value}} \tag{10}$$

4.4. Labeling method analysis

In this subsection, we give some analysis and experiments between our labeling approach and the traditional one. The traditional financial price prediction approaches like (Livieris et al., 2021; Zhang et al., 2022) commonly compare the current closing price p_t with the closing price $p_{t+\mathrm{w}}$ at time $t+\mathrm{w}$ to generate binary labels, as shown in (11). For each time t, label y_t is defined as 1 if the closing price of time $t + \mathrm{w}$ is larger than the current closing price, and $y_t = -1$ otherwise. We conduct subsequent comparison experiments for w taking the values 1,3,5,10 respectively.

$$y_t = \begin{cases} 1, \; if \;\; p_t < p_{t+\mathrm{w}} \\ -1, \; if \;\; p_t > p_{t+\mathrm{w}} \end{cases} \tag{11}$$

Fig. 5 gives a visual comparison of the generated labels between our labeling approaches and traditional labeling method with different w. First, our labeling method adds a new stable type, i.e. $y_t = 0$, to indicate that the currency price fluctuates randomly within a certain range during this period, thus avoiding unnecessary trades. Second, compared with the traditional method, our method obtains a relatively more stable label series, which effectively removes the influence of stochastic small-range price fluctuations and helps to drive the model to focus more on large fluctuations trend prediction. In addition, we also observe that the traditional approach leads to excessive observation of future information when w is too large, and thus erroneously over-passes forward the succeeding fluctuations caused by unexpected events at one moment.

Table 2 gives a parameter analysis result of θ_c and θ_s of our labeling approach, in which we adjust these two hyperparameters and test with our ACLMC model. We also counted the approximate proportion of the number of samples with labels −1, 0, and 1, which is significantly affected by these parameters. When fixing θ_c and changing θ_s from 0.045 to 0.09, the proportion sharply changes from 2.52:1:3.17 to 0.37:1:0.56, and the same thing happens when fixing θ_s and adjusting θ_c. We believe that this is in accordance with the definition of these two hyperparameters. When increasing θ_s, i.e. the threshold of stable trend, more intervals are judged to be stable intervals and allocate label 0, and when increasing θ_c, i.e. the trend change threshold, the trend will be harder to interrupt, leading to increasing the expected length of trend intervals, increasing the volatility within an interval, and thus

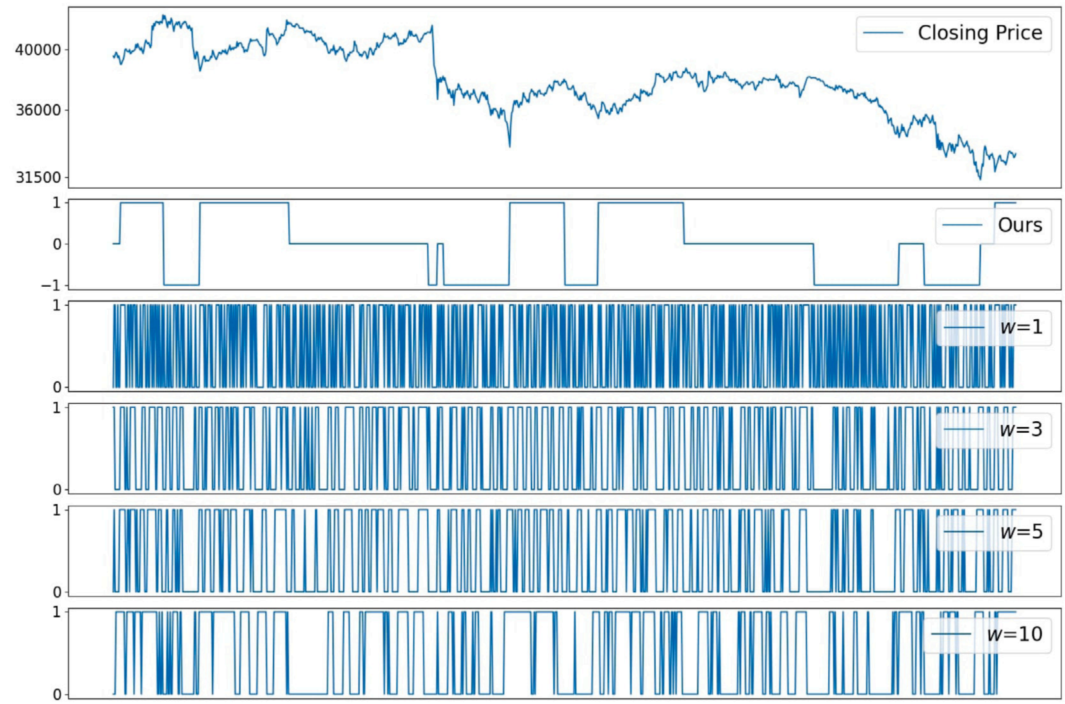


Fig. 5. Generated label series from our labeling method and traditional methods with different w. The 5 min data of BTC between 2021-05-20 and 2021-05-24 was randomly selected for plotting. From the top down are the closing price series, label series from our labeling method, and label series from traditional methods with w= 1, 3, 5, 10 respectively.

Table 2
Parameter analysis of our labeling approach. Proportion represents the approximate proportion of the number of samples with labels −1, 0, and 1 in order.

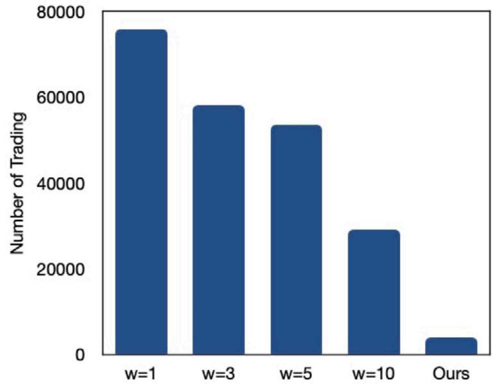
θ_c	θ_s	Proportion	Buy-hold				Long-short			
			ER (%)	NT	SR	MDD (%)	ER (%)	NT	SR	MDD (%)
0.03	0.045	2.52 : 1 : 3.17	24.08	5252	−1.53	−40.92	18.90	5252	−1.35	−53.19
	0.06	0.99 : 1 : 1.26	**43.00**	3927	**−0.51**	**−18.90**	**88.24**	3927	**1.14**	−36.65
	0.075	0.51 : 1 : 0.69	36.63	3207	−0.86	−28.59	61.97	3207	0.45	−32.69
	0.09	0.37 : 1 : 0.56	37.42	**1646**	−0.83	−27.48	52.16	**1646**	0.07	**−27.04**
0.02	0.06	0.36 : 1 : 0.51	32.88	**2146**	−1.11	−32.65	35.01	**2146**	−0.53	−48.54
0.03		0.99 : 1 : 1.26	**43.00**	3927	**−0.51**	**−18.90**	**88.24**	3927	**1.14**	**−36.65**
0.04		2.28 : 1 : 2.79	37.67	5762	−0.88	−29.96	48.53	5762	0.08	−55.02
0.05		6.24 : 1 : 7.52	31.94	4273	−1.03	−34.47	51.24	4273	0.03	−44.00

more likely to break the stable threshold and be assigned with label 1 or −1.

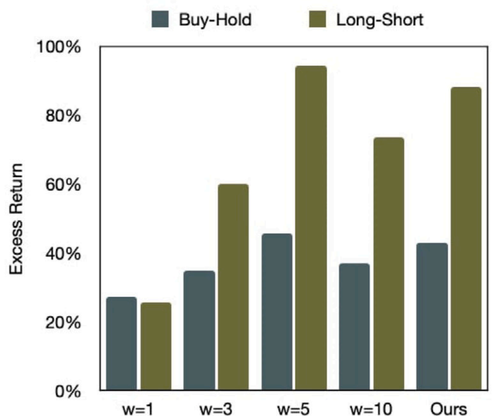
Due to the change in the proportion, the performance of the model obtained in several financial metrics also changes. When the proportion changes from a more balanced 0.99:1:1.26 to 0.37:1:0.56 with too many samples labeled as 0, the model tends to predict the current trend as stable due to unbalanced training data classes, thus making the number of transactions significantly lower and therefore missing trading opportunities, resulting in lower returns. When the number of samples with label 1 or −1 is higher, e.g., when the proportion is 2.52:1:3.17, the number of transactions is larger and the wrong prediction result is amplified in too many trades, leading to lower excess returns. Therefore, we believe that our labeling approach is sensitive to these two hyperparameters, which largely affect the proportion of the number of samples with different labels, influencing the model's tendency and ultimately the excess return and other financial indicators. Fortunately, the relationship between these two hyperparameters and the proportion is significant and highly correlated, so it is possible to quickly find appropriate setting values. We propose to statistics the proportion when generating labels and search for hyperparameters with a more balanced proportion. In this work, we set $\theta_c = 0.03$ and $\theta_s = 0.06$.

To demonstrate the superiority of our labeling methods, we train and backtest our ACLMC model with these five different methods and compare the number of trades and excess returns separately. The results are shown in Fig. 6. As can be seen, our method is slightly lower than the traditional labeling method in terms of excess returns, but this is in line with expectations. Our labeling method allows for short-term fluctuations under long-term trends, resulting in smoother labeling, while the traditional method focuses on all trend changes and is therefore more sensitive to ups and downs and more likely to achieve higher returns. However, this sensitivity also brings a serious increase in the number of transactions, which is tens of times higher for the traditional method than ours, and the resulting transaction overhead is unacceptable.

Moreover, to prove that our labeling method achieves a significant reduction in the number of transactions in different models, we compare the performance of using different labeling approaches under multiple deep learning methods, and the results are shown in Table 3. Multi-layer perceptron(MLP), CNN, LSTM, gated recurrent unit(GRU), attention-LSTM(ALSTM) and ACLMC are used. Clearly, regardless of the model used, our approach effectively helps reduce the number of trades by about 90% compared to traditional trend labels, avoiding expensive transaction overhead.



(a) Number of Transactions Comparison Histogram



(b) Excess Return Comparison Histogram.

Fig. 6. Comparison graph of the number of transactions and excess returns. The ACLMC is used for training and backtesting. The horizontal axis shows the traditional labeling method and our trend labeling method for 4 different w values, respectively.

Table 3
Number of transactions comparison of traditional labeling methods with our approach under multiple deep learning methods.

	MLP	CNN	LSTM	GRU	ALSTM	ACLMC
w = 1	53 091	65 845	89 620	94 907	89 690	75 811
w = 3	37 255	56 622	69 670	69 523	67 112	58 203
w = 5	40 499	56 580	59 700	58 989	55 422	53 504
w = 10	28 344	44 619	31 826	34 368	40 363	29 081
Ours	**4185**	**6533**	**3266**	**3680**	**4375**	**3927**

4.5. Model analysis

In this subsection, we will experimentally prove the effectiveness of the proposed ACLMC. We first compare our model with traditional machine learning methods such as SVM, RandomForests(RF), XGBoost, and deep learning methods like MLP, CNN, LSTM, GRU, and ALSTM. After that, we conducted ablation experiments to demonstrate the effectiveness of fusing data of different frequencies and different currencies through the attention mechanism.

Table 4 shows the average result of multiple tests comparing our model with various machine learning methods, and Fig. 7 plots the curves of excess returns obtained with these methods. Our trend-labeling methodology is used as the predictive target. Firstly, we find that CNN can focus more on obtaining local features and achieve better revenue results than networks with strong temporal information extraction such as LSTM and GRU, but also lead to a higher number of transactions. Meanwhile, ALSTM achieves better performance than LSTM, which shows the effectiveness of the attention module. Therefore, we believe that combining CNN, Attention module, and LSTM can integrate the advantages of these modules to optimize the performance of the model on multiple financial metrics. As expected, experiments prove that our model reaches the best investment performance among the tested methods. For both buy-hold and long-short strategies, our model achieved significantly better performance, where the excess return and Sharpe ratio are improved greatly. Meanwhile, the number of transactions remained within an acceptable range.

Table 4
Model comparison result between our model and various machine learning models.

Strategy	Model	ER (%)	NT	SR	MDD (%)
Buy-hold	MLP	27.36	4185	−1.39	−36.29
	CNN	31.40	6533	−1.43	−30.76
	LSTM	24.89	3266	−1.46	−37.62
	GRU	22.47	3680	−1.60	−40.29
	ALSTM	30.42	4375	−1.38	−32.01
	SVM	34.22	4669	−0.97	−33.98
	RF	19.65	4332	−1.72	−45.56
	XGBoost	24.42	**1457**	−1.36	−40.25
	ACLMC(Ours)	**43.00**	3927	**−0.51**	**−18.90**
Long-short	MLP	31.76	4185	−0.68	−40.95
	CNN	43.43	6533	−0.11	−34.82
	LSTM	20.67	3266	−1.47	−42.73
	GRU	19.74	3680	−1.53	−43.83
	ALSTM	40.11	4375	−0.35	−31.75
	SVM	54.67	4669	0.16	**−25.25**
	RF	14.12	4332	−1.93	−51.74
	XGBoost	25.80	**1457**	−1.25	−39.63
	ACLMC(Ours)	**88.24**	3927	**1.14**	−36.65

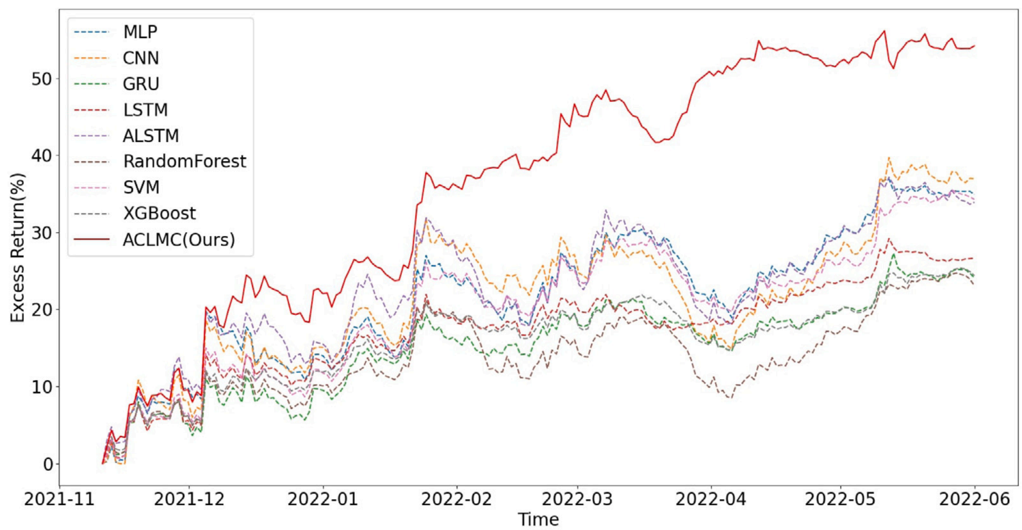
Table 5
Model ablation experiments result. CAL represents the currencies attention layer and FAL represents the frequencies attention layer.

Strategy	CAL	FAL	ER (%)	NT	SR	MDD (%)
Buy-hold			28.45	4499	−1.35	−33.51
	✓		31.09	**3884**	−1.17	−35.61
		✓	30.75	4457	−1.30	−32.48
	✓	✓	**43.00**	3927	**−0.51**	**−18.90**
Long-short			36.50	4499	−0.70	−38.43
	✓		40.05	**3884**	−0.45	−43.50
		✓	46.09	4457	−0.34	**−32.38**
	✓	✓	**88.24**	3956	**1.14**	−36.65

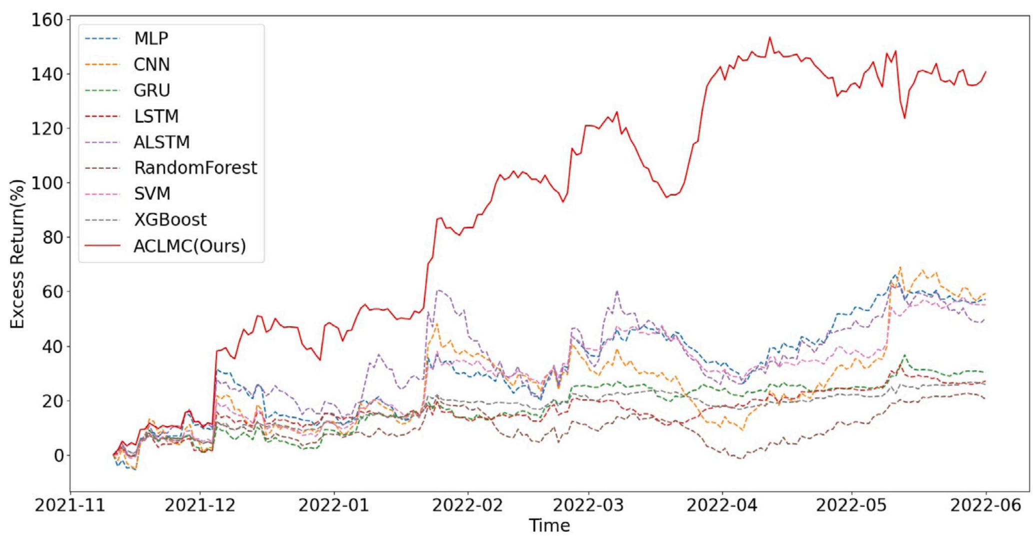
Table 5 shows the ablation experiment results, where we prove the effectiveness of the designed attention modules in our model. CAL in the table represents the currency attention layer and FAL represents the frequency attention layer. It is clear that both these modules have respectively improved financial metrics, such as Sharpe ratio and maximum drawdown, and have achieved an excess return growth of about 4%. Further, the excess returns with employing these two modules simultaneously doubled compared to not adding them, the SR and MDD were significantly optimized, and the number of transactions remained at acceptable levels. Therefore, it can be concluded that our inclusion of these two modules is effective in optimizing the model performance and improving the corresponding strategy returns.

Further, we conducted an experimental analysis of the multi-currency combination approach, which actually includes both input and output components, i.e., training the network using multi-currency data and using the multi-currency simultaneous output for trading.

Table 6 shows the effectiveness of adding multiple currencies for training. For the selected 5 coins BTC, ETH, BNB, XRP, and ADA, we tested the model training using data from 1 to 5 coins, using only the prediction results of BTC for single coin transactions and statistical financial indicators. Experiments show that adding more currency data for training can effectively promote the model to observe more data distributions and improve the predictive power of the model. By adding data from the other four currencies to assist in training, the model



(a) Buy-Hold Strategy Excess Return Curve.



(b) Long-Short Strategy Excess Return Curve.

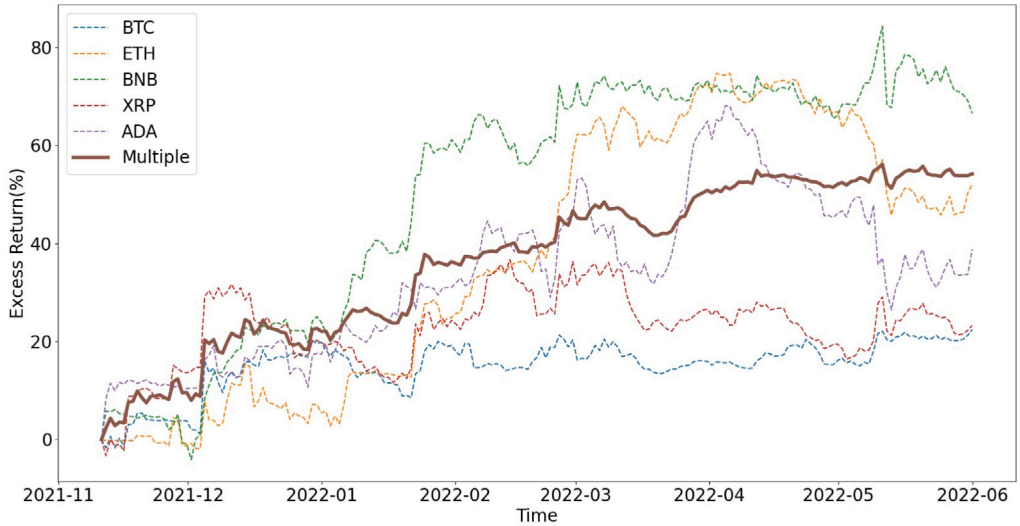
Fig. 7. Excess return curves for our approach and other machine learning approaches under the buy-hold strategy and long-short strategy. The dashed line indicates machine learning methods used for comparison, and the solid line indicates our ACLMC model. The best results from multiple tests were selected to plot for each model and sampling was performed and plotted every 300 points/1500 min for better visualization.

significantly improved the excess return on BTC trades from 6.05% to 15.40%, and the Sharpe ratio and maximum drawdown were also significantly optimized.

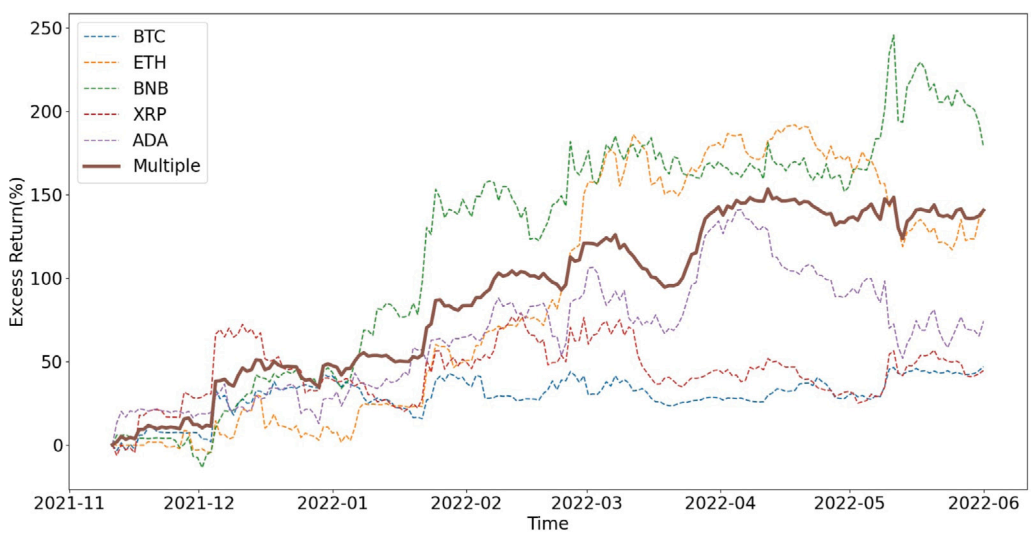
Fig. 8 compares the simulated trading excess return curves for forecasting 5 currencies simultaneously and for a single currency only to demonstrate the benefits of simultaneous forecasting for multiple currencies. We train the model using all five currencies but predict and simulate trading only for given currencies. Obviously, predicting and investing in a single currency may achieve higher returns, but it also entails greater drawdown risk. This is because the current methods all struggle to obtain a relatively low level of accuracy, and the model's performance in predicting one currency is inconsistent, making all funds fully dependent on the model's prediction of one currency subject to large losses in case of error. Our model supports integrated multi-currency investments by outputting multi-currency forecasts simultaneously, spreading the investment risk, and obtaining a smoother return curve, as shown by the solid line in the figure.

Table 6
Using different amounts of cryptocurrency data to train the model but simulate trading on BTC only. The c coins represent training using the top c cryptocurrency data from BTC, ETH, BNB, XRP, and ADA.

Strategy	Coins	ER (%)	NT	SR	MDD (%)
Buy-hold	1	6.05	**583**	−2.05	−54.96
	2	1.57	618	−2.34	−58.58
	3	8.56	707	−1.81	−53.62
	4	14.39	622	**−1.32**	−49.97
	5	**15.40**	635	−1.40	**−47.89**
Long-short	1	7.72	**583**	−1.37	−65.02
	2	−1.24	618	−1.82	−69.28
	3	14.53	707	−1.13	−56.65
	4	27.83	622	−0.49	−48.53
	5	**28.53**	635	**−0.45**	**−47.69**



(a) Buy-Hold Strategy Excess Return Curve.



(b) Long-Short Strategy Excess Return Curve.

Fig. 8. Excess return curves for simultaneous forecasting of multiple currencies versus investment simulations with only a single currency. The dashed line indicates investing in these 5 currencies alone, and the solid line indicates forecasting and investing in all 5 currencies at the same time. The best results from multiple tests were selected to plot for each model and sampling was performed and plotted every 300 points/1500 min for better visualization.

Table 7
Using different amounts of cryptocurrency data to train the model and simulate trading. The c coins represent training and testing using the top c cryptocurrency data from BTC, ETH, BNB, XRP, and ADA.

Strategy	Coins	ER (%)	ANT	SR	MDD (%)
Buy-hold	1	6.05	**583**	−2.05	−54.96
	2	15.91	683	−2.24	−45.67
	3	25.48	711	−1.97	−41.20
	4	33.34	755	−1.37	−36.48
	5	**43.00**	785	**−0.51**	**−18.90**
Long-short	1	7.72	**583**	−1.37	−65.02
	2	12.07	683	−1.45	−58.87
	3	35.81	711	−0.82	−50.59
	4	52.22	755	−0.09	−44.44
	5	**88.24**	785	**1.14**	**−36.65**

Table 7 combines multiple inputs and outputs for experiments, e.g., when using 3 coins, data for BTC, ETH, and BNB are simultaneously input and predicted for simulated trading, thus further demonstrating that the multiple-input–output model structure we have adopted contributes to the improvement of financial metrics for investment via trend prediction. For better comparison, the average number of transactions per currency(ANT) is used here. The results show that both ER and MDD increase with the number of currencies adopted. This is exactly the combination of the advantages of each of the above two parts. Multi-currency provides more training data with relevant information, which helps improve model performance. The MDD is mitigated by the fact that multi-currency investments can smooth out the losses caused by single-currency forecast errors. In addition to this, ANT rises with the number of currencies, which is the result of different currencies having different numbers of local minimum, bringing more trading opportunities. Moreover, the increase in ANT is relatively small, so it is acceptable to sacrifice a certain number of transactions in exchange for a smoother and more profitable return.

5. Conclusion

In this paper, we propose a new labeling method for high-frequency trading that can reduce the interference of the randomness of financial price series on label generation, thus obtaining more stable trend labels, implicitly enabling the model to generate smoother prediction series, and effectively reducing the unacceptable excessive number of trades brought about by traditional methods under high-frequency trading. We also propose a novel ACLMC model that can combine financial factors with different statistical frequencies and different monetary information using an attention mechanism to effectively improve the performance of the model and to improve and optimize various financial indicators. Multi-currency trading is supported by simultaneous forecasts in multiple coins, smoothing out investment risks caused by forecast errors in a single currency. Experimentally, our labeling approach proves a significantly lower number of transactions with less excess return reduction compared to traditional approaches. Our model also significantly improves various financial metrics when comparing multiple machine learning and deep learning baseline models.

In addition, we also note some limitations of this work. On the one hand, the labeling method is sensitive to hyperparameters, and inappropriate hyperparameters may lead to class imbalance and mislead the forecasting model. On the other hand, in such a highly volatile market, a 5-min trading frequency may still not be sufficient for black swan events that are beyond normal expectations. Future work will be devoted to finding an automated parameter tuning algorithm to further optimize the labeling method, or to improve the stability of the model to cope with class-imbalance data, thus further improving the performance of the method and further considering the possibility of higher trading frequencies. It is also worth considering how to evaluate the impact of unforeseen events and incorporate them into our approach.

CRediT authorship contribution statement

Peng Peng: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Writing – review & editing. **Yuehong Chen:** Investigation, Formal analysis, Resources, Data curation. **Weiwei Lin:** Validation, Supervision, Project administration, Writing – review & editing. **James Z. Wang:** Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by National Natural Science Foundation of China (62072187, 62002078), Guangdong Major Project of Basic and Applied Basic Research (2019B030302002), Guangdong Marine Economic Development Special Fund Project (GDNRC[2022]17), Guangzhou Development Zone Science and Technology Project (2021GH10) and in part by The Major Key Project of PCL, China (PCL2023AS7-1). James Z. Wang's work was supported (in part) by the National Institutes of Health (NIH)/Eunice Kennedy Shriver National Institute of Child Health and Human Development (R01 HD 069374) and the National Science Foundation (NSF) DBI (1759856).

References

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*, 53.

Ding, G., & Qin, L. (2020). Study on the prediction of stock price based on the associated network model of LSTM. *International Journal of Machine Learning and Cybernetics*, *11*(6), 1307–1317.

Dutta, A., Kumar, S., & Basu, M. (2020). A gated recurrent unit approach to bitcoin price prediction. *Journal of Risk and Financial Management*, *13*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*, 1735–1780.

Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, *4*, 1–30.

Kingma, D. P., & Ba, J. L. (2014). Adam: A method for stochastic optimization. In *3rd international conference on learning representations, ICLR 2015 - Conference track proceedings*.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Li, H., Shen, Y., & Zhu, Y. (2018). Stock price prediction using attention-based multi-input LSTM. *Proceedings of Machine Learning Research*, *95*, 454–469.

Li, X., & Wu, P. (2022). Stock price prediction incorporating market style clustering. *Cognitive Computation*, *14*(1), 149–166.

Liang, X., Ge, Z., Sun, L., He, M., & Chen, H. (2019). LSTM with wavelet transform based data preprocessing for stock price prediction. *Mathematical Problems in Engineering*, *2019*, Article 1340174.

Livieris, I. E., Kiriakidou, N., Stavroyiannis, S., & Pintelas, P. (2021). An advanced CNN-LSTM model for cryptocurrency forecasting. *Electronics*, *10*.

Loshchilov, I., & Hutter, F. (2016). SGDR: Stochastic gradient descent with warm restarts. In *5th international conference on learning representations, ICLR 2017 - Conference track proceedings*.

Lu, W., Li, J., Wang, J., & Qin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. *Neural Computing and Applications*, *33*, 4741–4753.

Ma, Y., Ventre, C., & Polukarov, M. (2022). Denoised labels for financial time series data via self-supervised learning. In *Proceedings of the third ACM international conference on AI in finance* ICAIF '22, (pp. 471–479). New York, NY, USA: Association for Computing Machinery.

Matsubara, T., Akita, R., & Uehara, K. (2018). Stock price prediction by deep neural generative model of news articles. *IEICE Transactions on Information and Systems*, *101*(4), 901–908.

Mohan, S., Mullapudi, S., Sammeta, S., Vijayvergia, P., & Anastasiu, D. C. (2019). Stock price prediction using news sentiment analysis. In *2019 IEEE fifth international conference on big data computing service and applications (BigDataService)* (pp. 205–208).

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, 21260.

Ozer, F., & Okan Sakar, C. (2022). An automated cryptocurrency trading system based on the detection of unusual price movements with a Time-Series Clustering-Based approach. *Expert Systems with Applications*, *200*, Article 117017.

Rezaei, H., Faaljou, H., & Mansourfar, G. (2021). Stock price prediction using deep learning and frequency decomposition. *Expert Systems with Applications*, *169*, Article 114332.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2017). Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017 international conference on advances in computing, communications and informatics (ICACCI)* (pp. 1643–1647).

Sezer, O. B., & Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, *70*, 525–538.

Shintate, T., & Pichl, L. (2019). Trend prediction classification for high frequency bitcoin time series with deep learning. *Journal of Risk and Financial Management*, *12*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Vidal, A., & Kristjanpoller, W. (2020). Gold volatility prediction using a CNN-LSTM approach. *Expert Systems with Applications*, *157*, Article 113481.

Wang, Y., Huang, M., Zhu, X., & Zhao, L. (2016). Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing* (pp. 606–615).

Wu, J. M.-T., Li, Z., Herencsar, N., Vo, B., & Lin, J. C.-W. (2021). A graph-based CNN-LSTM stock price prediction algorithm with leading indicators. *Multimedia Systems*.

Wu, D., Wang, X., Su, J., Tang, B., & Wu, S. (2020). A labeling method for financial time series prediction based on trends. *Entropy*, *22*, 1162.

Yıldırım, D. C., Toroslu, I. H., & Fiore, U. (2021). Forecasting directional movement of Forex data using LSTM with technical and macroeconomic indicators. *Financial Innovation*, *7*, 1.

Zhang, Z., Dai, H. N., Zhou, J., Mondal, S. K., García, M. M., & Wang, H. (2021). Forecasting cryptocurrency price using convolutional neural networks with weighted and attentive memory channels. *Expert Systems with Applications*, *183*, Article 115378.

Zhang, Q., Qin, C., Zhang, Y., Bao, F., Zhang, C., & Liu, P. (2022). Transformer-based attention network for stock movement prediction. *Expert Systems with Applications*, *202*, Article 117239.

Zhao, Z., Rao, R., Tu, S., & Shi, J. (2018). Time-weighted lstm model with redefined labeling for stock trend prediction. In *Proceedings - International conference on tools with artificial intelligence, ICTAI, Vol. 2017-November* (pp. 1210–1217). IEEE Computer Society.

Zhao, W., Wu, F., Fu, Z., Wang, Z., & Zhang, X. (2020). Sentiment analysis on weibo platform for stock prediction. In *International conference on artificial intelligence and security* (pp. 323–333). Springer.