GEORGE BROWN PRESS
JOURNALS, BOOKS & PROCEEDINGS

Journal of Computer, Signal, and System Research

Article

Research on China's Stock Index Futures Pair Trading Strategy Based on High-Frequency Data

Jun Dong [1,*]

[1] Shenyang Yingqi Technology Co., Ltd, Shenyang, Liaoning, China
* Correspondence: Jun Dong, Shenyang Yingqi Technology Co., Ltd, Shenyang, Liaoning, China

Abstract: The efficiency and accuracy of high-frequency trading strategies are increasingly emphasized in modern financial markets, especially in China's stock index futures market. Bayesian theory and probabilistic neural network algorithms have become important tools for constructing high-frequency trading models because of their powerful predictive ability and adaptivity. By studying the minimum error rate and minimum risk Bayesian decision in Bayesian theory, as well as the structure and training methods of probabilistic neural networks, this thesis aims to develop a pair trading strategy for stock index futures based on high-frequency data. The division of dataset and the classification method of classically unbalanced dataset are the key steps of the research, through which the performance of the model is optimized and the effectiveness of the strategy is verified through back-testing experiments. This study not only improves the accuracy and stability of the high-frequency trading strategy, but also provides a new idea and method for quantitative trading in China's financial market.

Keywords: high-frequency data; stock index futures; paired trading strategies

1. Introduction

The rapid development of China's stock index futures market has made high-frequency trading strategies the focus of investors' attention. Bayesian theory and probabilistic neural network algorithms have gradually become important methods for constructing high-frequency trading models by virtue of their superior performance in dealing with complex data and uncertainty problems. Bayes theorem updates the prior knowledge of the model through conditional probability, and minimum error rate Bayesian discrimination improves the predictive accuracy of the model by maximizing the probability of correct classification. And the minimum risk Bayesian decision not only pursues correct classification, but also considers the risk cost of different decisions. Probabilistic neural network, as a kind of nonlinear classifier, can effectively deal with complex pattern recognition problems. Combining these theories and algorithms, this thesis constructs a pair trading strategy for stock index futures based on high-frequency data, aiming to improve the performance of the strategy through the optimization of the dataset and the comparison of the models, and to provide a reference for practical applications.

2. A Study of Bayesian Theory and Probabilistic Neural Network Algorithms

2.1. Bayes' Theorem

Bayes' Theorem is a fundamental principle in probability theory used to update prior probabilities as new evidence emerges. This principle is particularly important in high-frequency trading in the financial markets, as it helps traders adjust their strategies based on real-time data to better cope with market uncertainty. The Bayesian algorithm, on the other hand, is a statistical method based on Bayes' theorem that is widely used in forecasting and decision making in financial markets. The algorithm updates the parameters of

Received: 8 February 2025
Revised: 13 February 2025
Accepted: 27 February 2025
Published: 28 February 2025

Copyright: © 2025 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

the model by combining prior knowledge and new evidence to maintain high forecasting accuracy in a changing market environment [1]. In high-frequency trading, the Bayesian algorithm is able to efficiently process a large amount of real-time data and capture short-term fluctuations and trends in the market by dynamically adjusting the model parameters. Its principle is shown in Figure 1:

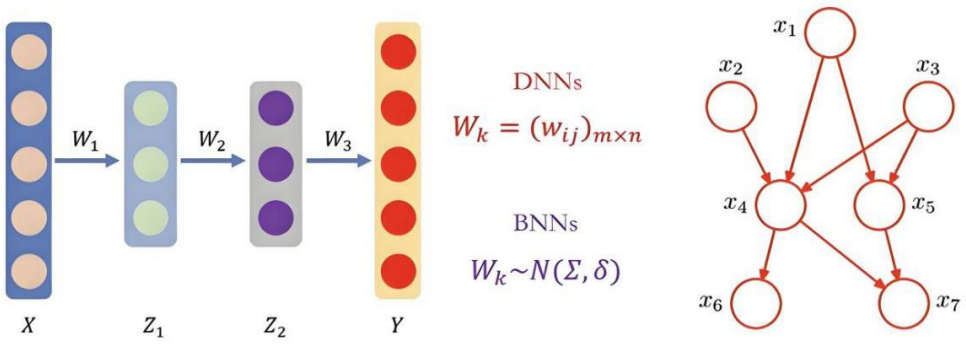

Figure 1. Bayesian algorithm.

Bayes' theorem is formally expressed as in equation (1):

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)} \tag{1}$$

Where P(A|B) denotes the posterior probability of event A occurring under the condition of event B occurring, P(B|A) is the likelihood probability of event B occurring under the condition of event A occurring, P(A) is the prior probability of event A and P(B) is the marginal probability of event B. Through Bayes' theorem, the parameters of the trading model can be dynamically adjusted in the process of constant changes in market data, thus improving the accuracy of prediction.

The core of Bayes' theorem lies in the calculation of conditional probability. In high-frequency trading, conditional probabilities can be used to assess the impact of specific market signals on stock price movements. For example, assuming that event A indicates that the price of a stock will rise, and event B indicates that a technical indicator sends a buy signal, then P(A|B) can represent the probability that the stock price will rise in the event that the technical indicator sends a buy signal [2]. The calculation of this probability is based on historical data, and the predictive ability of the model can be gradually improved by continuously updating the prior probability and likelihood probability.

Bayesian decision theory further extends the application of Bayes' theorem. Minimum error rate Bayesian discrimination is an important concept in classification problems in financial markets. Minimum error rate Bayesian discrimination selects the optimal decision by maximizing the probability of correct classification. Specifically, for two categories C1 and C2, assuming x is an observed data point, the decision rule for the minimum error rate Bayesian discrimination is as in Equation (2):

$$IF \quad P(C1 \mid x) > P(C2 \mid x), choose \quad C1; otherwise, chooseC2 \tag{2}$$

This rule is particularly important in high-frequency trading because it helps traders make more accurate decisions in highly volatile and rapidly changing market environments. By comparing the posterior probabilities of different classes under a given observed data point, the Minimum Error Rate Bayesian Discriminant effectively reduces the risk of misclassification and thus improves the robustness of trading strategies [3].

Minimum risk Bayesian decision-making not only considers the correctness of classification, but also the risk cost of decision-making. In the financial market, different trading decisions may bring different benefits and risks. Minimum risk Bayesian decision making measures the risk of a decision by introducing a loss function L, which is usually

defined as the loss incurred in case of misclassification. The decision rule for minimum risk Bayesian decision making is shown in Equation (3):

$$g(x) = \arg_k^{\min} \sum_{i=1}^{K} L(k \mid i) \cdot P(i \mid x) \tag{3}$$

Where $g(x)$ denotes the category selected under the observed data point x, L(k|i) is the loss of discriminating the category as k when the actual category is i, and P(i|x) is the posterior probability of category i under the observed data point x. By minimizing the risk, the Minimum Risk Bayesian Decision balances the return and risk, resulting in better performance in real trading.

2.2. Probabilistic Neural Network

Probabilistic neural network (PNN) is a neural network model based on Bayesian theory, which is widely used in classification and regression tasks, as shown in Figure 2. Different from traditional neural networks, PNN smoothes the input data by Gaussian kernel function, thus showing high accuracy and stability in classification tasks.The structure of PNN consists of an input layer, a pattern layer, a summation layer, and an output layer, where each node in the pattern layer corresponds to a training sample, and the probability of each category can be obtained by calculating the similarity between the input data and the training samples. This structure enables the PNN to effectively deal with noise and outliers in high-frequency data and improve the robustness of the model [4].

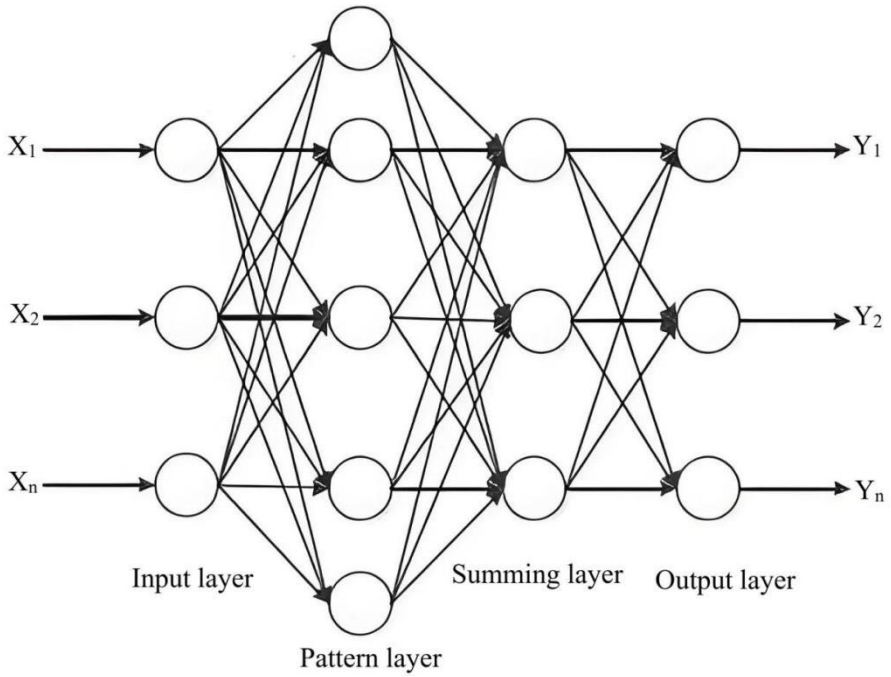

Figure 2. Probabilistic Neural Network (PNN).

The application of PNN is particularly significant in high-frequency trading in financial markets. High-frequency trading data is characterized by short time intervals and high volatility, which puts high demands on the model's real-time performance and accuracy.PNN, by using Gaussian kernel function, is able to quickly capture the changes of the market signals, while maintaining the smoothness of the model and avoiding over-fitting. In addition, PNN excels in handling multi-featured inputs and is able to combine multiple market indicators and historical data to provide a more integrated and comprehensive classification result. This approach is not only effective in the stock market, but also widely used in the foreign exchange market, futures market and other financial fields [5].

2.2.1. Parzen Window Method

The Parzen window method, a non-parametric probability density estimation technique, is extensively employed in probabilistic neural networks (PNNs) to address intricate input data distributions. This approach estimates the probability density of input data by defining a smooth kernel function around sample points. The essence of the Parzen window method lies in the judicious selection of the kernel function and bandwidth, which directly influences the accuracy and robustness of the probability density estimation. In practical applications, the Gaussian kernel function is predominantly favored due to its smoothness and computational simplicity, enabling effective handling of voluminous data in high-frequency trading.

In the realm of high-frequency trading within financial markets, the Parzen window method facilitates the rapid detection of market signal variations by probabilistic neural networks, concurrently mitigating the impact of data noise on the model. High-frequency trading data is characterized by short time intervals and significant volatility, imposing stringent demands on the model's real-time performance and accuracy. By constructing a Gaussian kernel function around each training sample, the Parzen window method maps input data points to the probability density distribution of sample points, enabling PNNs to make precise classification decisions swiftly. This methodology is not only applicable to stock markets but also holds significant potential in foreign exchange, futures, and other financial domains [6].

The Parzen window method excels particularly in handling multi-feature inputs. By mapping multiple features to a multi-dimensional probability density distribution, PNNs can synthesize various market indicators and historical data, yielding more comprehensive and holistic classification outcomes. Furthermore, the non-parametric nature of the Parzen window method allows it to adapt to data distributions under varying market conditions, circumventing the overfitting issues inherent in traditional parametric methods when data distributions shift. This adaptability endows PNNs with substantial practical value and broad application prospects in the dynamic and complex financial markets.

2.2.2. Specific Algorithms for Probabilistic Neural Networks

Probabilistic neural network (PNN) is an efficient classification algorithm based on Bayesian theory.The specific algorithm consists of four steps: normalization, calculating the pattern distance, activating the neurons of radial basis function of the sample layer, and setting the number of samples.The structure of the PNN consists of an input layer, a pattern layer, a summation layer, and an output layer. The input layer receives the sample data x to be classified and normalizes it to eliminate the difference in magnitude between different features. The normalization formula is (4):

$$\mathrm{X}_{norm} = \frac{x - \mu}{\sigma} \tag{4}$$

Where μ and σ are the mean and standard deviation of the training data, respectively.

Each node in the pattern layer corresponds to a training sample x_i, and their similarity is determined by calculating the pattern distance between the input samples and the training samples. The formula for calculating the pattern distance is (5):

$$d_i = \left\| x_{norm} - x_{i,norm} \right\| \tag{5}$$

Where $\left\| \cdot \right\|$ denotes the Euclidean distance. The result of the pattern distance calculation is used to activate the Radial Basis Function (RBF) neurons in the pattern layer.The RBF is usually a Gaussian kernel function with the activation formula in (6):

$$\mathrm{K}(d_i) = \exp(-\frac{d_i^2}{2\sigma^2}) \tag{6}$$

The window width parameter σ controls the degree of smoothing of the Gaussian kernel function and has a significant impact on model performance.

The summation layer summarizes the outputs of the activated RBF neurons in the pattern layer and calculates the total similarity of each category. Let the number of training samples for category j be n_j, then the total similarity S_j of category j is (7):

$$S_j = \sum_{i=1}^{n_j} K(d_i) \tag{7}$$

This process yields the total similarity of each category by summing the similarity of all training samples within that category. The output of the summation layer is the total similarity of each category.

The output layer calculates the probability of each category based on the result of the summation layer. The probability P_j of category j is (8):

$$P_j = \frac{S_j}{\sum_{k=1}^{C} S_k} \tag{8}$$

where C is the total number of categories. The probability calculations are normalized to ensure that the sum of the probabilities of all categories is 1, which facilitates the final classification decision.

3. Pattern Recognition Model Based on High Frequency Data

3.1. Data Set Segmentation

Dataset division is an important step in machine learning in order to train and test the validity of the model. In this paper, the data for stock index futures from January 5 to January 9, 2016 are divided in detail. Specifically, the data from January 5 to January 8 is divided into a training set and a testing set, where 70% of the data is used as the training set and the remaining 30% is used as the testing set.The data from January 9 is used exclusively for the testing set in order to evaluate the performance of the model on unknown data [7]. This division ensures the independence of the training and testing processes, while also providing a set of data without any training information for the final testing of the model. The up / down distribution of the training set is shown in Table 1.

Table 1. distribution of ups and downs in the training set.

Rise and fall distribution	Frequency	Scale
flat	51,590	60.16%
rose	17,144	19.99%
fall	17,016	19.84%

The data volume of the training set, test set and test set are 85,750, 36,750 and 31,600 samples, respectively. Further analyzing the up and down distribution of the training set, it can be found that the number of flat samples is 51,590, accounting for 60.16% of the total number of samples in the training set; the number of up samples is 17,144, accounting for 19.99%; and the number of down samples is 17,016, accounting for 19.84%. This distribution feature indicates that the number of flat samples in the training set is significantly larger than the number of up and down samples, forming a typical category imbalance dataset. The category imbalance problem is a common challenge in pattern recognition, and if such data are used directly for training, the model may tend to predict most categories, which leads to biased classification results.

3.2. Classification Methods for Category Imbalanced Datasets

3.2.1. Resampling Methods

The resampling method is a commonly used classification strategy when dealing with category unbalanced datasets. Random oversampling increases the number of samples from a few classes by randomly selecting and replicating them to improve the balance

of the dataset. Although this method is able to increase the number of samples from a few classes, it does not provide new information and therefore may lead to an increase in the training time of the classifier. Especially in the case of more noise in the dataset, random oversampling is prone to trigger the overfitting problem, which makes the classifier generate the same minority class rules, thus affecting the generalization ability of the model [8]. The principle of the resampling method is shown in Figure 3.

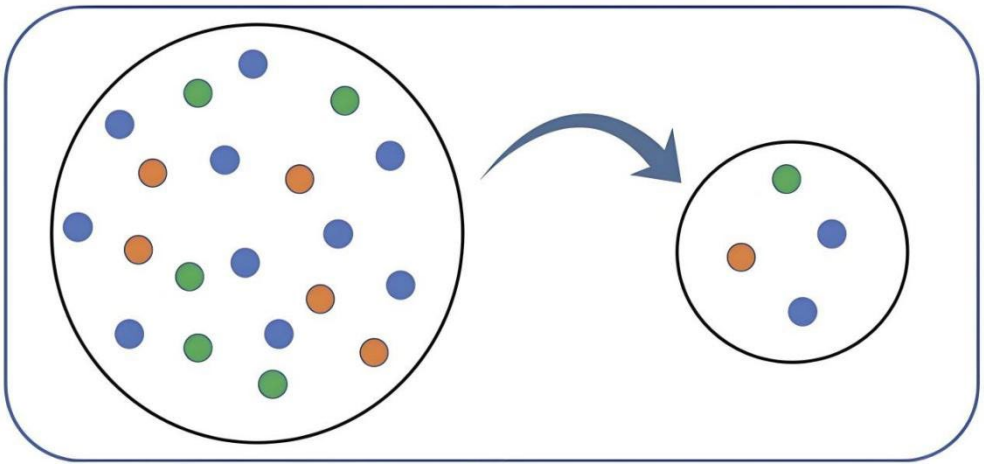

Figure 3. Principle of the resampling method.

To overcome these limitations, researchers have proposed several improved resampling methods. One of them is the Synthetic Minority Over-Sampling Technique (SMOTE), which works by synthesizing new samples in the feature space of the minority samples instead of simply copying the existing samples.The SMOTE method is able to provide more information to the model while keeping the dataset balanced, reducing the risk of overfitting.

3.2.2. Under-Sampling Method

The undersampling method is a strategy to increase the degree of balancing of the dataset by removing samples from the majority class. Unlike oversampling methods, undersampling achieves balance by reducing the number of majority class samples, as shown in Figure 4. Random undersampling is a basic undersampling method that can quickly simplify the dataset by randomly removing some of the majority class samples. However, a major drawback of this method is that important potential classification information may be lost, thus affecting the performance and classification ability of the classifier.

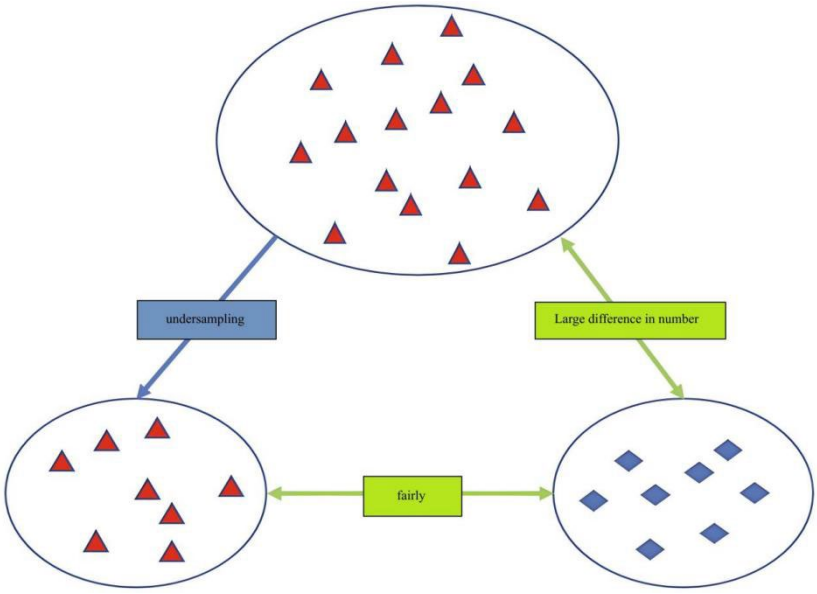

Figure 4. Under-sampling method.

In order to overcome the shortcomings of random undersampling, Tomek proposed the Tomek pairwise method in 1976. The basic principle of this method is to reduce the interference of majority class samples with classification boundaries by identifying and removing those majority class samples that are adjacent to minority class samples. The Tomek pairing method is able to retain the portion of the majority class samples that are distant from the minority class samples, thus reducing the information loss and improving the robustness of the classifier while maintaining a balanced dataset.

The undersampling method is important in pattern recognition of high-frequency data, especially when the number of majority class samples in the dataset far exceeds the number of minority class samples [9].

4. High-Frequency Trading Model Construction and Backtesting Experiments

4.1. Evaluation of High-Frequency Trading Strategy Performance

Performance evaluation of high frequency trading strategies involves a variety of metrics to ensure a comprehensive and scientific assessment. Yield is the core metric for evaluating a trading strategy and it reflects the profitability of the strategy. The rate of return is calculated as the profit of the trading strategy divided by its cost, but the rate of return itself can vary due to the effects of different time periods. In order to fairly compare the returns of different trading strategies, it is usually necessary to ensure that the length of time over which the return is calculated is consistent. In addition, the volatility of returns is an important evaluation metric that measures the degree of volatility of a strategy's returns, usually expressed as the standard deviation of the returns. The lower the volatility, the more stable the strategy is.

Maximum retracement is another key metric that measures the maximum loss over a trading time period. Maximum retracement helps traders understand how a strategy performs in unfavorable market conditions and assess its potential maximum risk. In addition to this, the Sharpe Ratio is one of the commonly used evaluation metrics that assesses the risk-adjusted return of a strategy by considering the difference between the strategy's rate of return and the risk-free rate of return, as well as its volatility. A higher Sharpe Ratio indicates that the strategy has a higher return for the same risk taken or a lower risk for the same return.

4.2. High Frequency Trading Strategy Construction

The high-frequency trading strategy model is constructed based on the probabilistic neural network model derived from the comparative analysis in Chapter IV. The specific steps of the strategy are as follows:

1) Feature Construction: At each moment, nine input features are constructed from the generated data, including bid-ask spread, sell-side depth, buy-side depth, difficulty of buy price change, price fluctuation range, percentage of upward movements, percentage of downward movements, buy-side order volume, and sell-side order volume. These features are designed to capture market dynamics and provide critical inputs for the model.
2) Model Inference: The input features from the previous 10 ticks (corresponding to 5 seconds) are used as the characteristics of a new sample and fed into the pre-trained probabilistic neural network. Based on the network's assessment, if the probability of an upward movement exceeds 0.8, it is classified as a rise; if the probability of a downward movement exceeds 0.8, it is classified as a fall; otherwise, it is classified as a sideways movement [10].
3) Action Based on Classification: If the classification is a rise, a long position is established at the next tick (corresponding to 0.5 seconds) at the bid price, and the position is closed by establishing a short position at the ask price after holding for 6 ticks.
4) Action Based on Classification: If the classification is a fall, a short position is established at the next tick at the ask price, and the position is closed by establishing a long position at the bid price after holding for 6 ticks (corresponding to 3 seconds). If the classification is a sideways movement, no trade is executed.

The strategy employs a time-based stop-loss and take-profit mechanism, meaning that positions are automatically closed upon reaching the holding time, with no other conditions triggering a close. Through these steps, the high-frequency trading strategy is capable of swiftly reacting to market fluctuations, thereby enhancing both the accuracy and profitability of transactions.

4.3. High Frequency Trading Backtest Results

In this paper, the Bayesian discriminant model and probabilistic neural network model are investigated using data from January 5 to January 8, 2016 In the high-frequency trading backtesting experiments, the data from January 5 to January 8, 2016 are used to train the probabilistic neural network model, and the data from January 9 is used for backtesting. The backtest results show that the high-frequency trading strategy based on the threshold-based probabilistic neural network has high profitability, but there is a large volatility in the returns. This may be related to the failure to set a stop-loss strategy, especially when the price falls continuously, the loss of the strategy is more significant. By comparing the cumulative returns with the price movements, it was found that the strategy's losses were higher when the price fell continuously, thus affecting the overall stability. In order to improve this problem, a stop-loss mechanism was added to the original strategy: when there is a continuous significant price decline or increase, no trading is done. When the loss reaches 0.2%, a stop loss is implemented at the next tick and the trade is withdrawn to the next trading cycle. The adjusted backtest results show that after adding the stop-loss strategy, the volatility of the strategy's return is significantly reduced and the overall performance is more stable. Figure 5 illustrates the backtest results without adding the stop-loss mechanism, showing that the strategy does have a higher rate of return in a given time period, but with higher volatility.

Figure 5. High-frequency trading strategy backtested returns.

Figure 6 shows the backtest results after adding the stop-loss mechanism, showing that the cumulative return curve of the strategy is much smoother, which significantly reduces the large losses due to consecutive downturns. These results show that a reasonable stop-loss strategy can effectively improve the robustness and profitability of the high-frequency trading strategy and make it perform more robustly in the complex and changing market environment. With these improvements, the performance of the high-frequency trading model is significantly enhanced, providing a stronger basis for practical applications.

Figure 6. High-frequency trading strategy improved return charts.

5. Conclusion

Through in-depth study of Bayesian theory and probabilistic neural network algorithm, this thesis successfully constructs a pair trading strategy for stock index futures based on high-frequency data. The reasonable division of the dataset and the effective treatment of the category imbalance problem significantly improve the classification performance of the model. Comparative experiments between the Bayesian discriminant model and the probabilistic neural network model further verify the superiority of the algorithm. The results of high-frequency trading backtesting show that the strategy performs well in actual trading, which not only improves the accuracy and stability of trading, but also provides new ideas for quantitative trading in China's financial market. Future research can further optimize the model parameters and introduce more market factors to enhance the generalization ability and practical application of the strategy.

References

1. J. Luo, Y. C. Lin, and S. Wang, "Intraday high-frequency pairs trading strategies for energy futures: Evidence from China," *Appl. Econ.*, vol. 55, no. 56, pp. 6646–6660, 2023, doi: 10.1080/00036846.2022.2161993.
2. C. He, T. Wang, X. Liu, *et al.*, "An innovative high-frequency statistical arbitrage in Chinese futures market," *J. Innov. Knowl.*, vol. 8, no. 4, p. 100429, 2023, doi: 10.1016/j.jik.2023.100429.
3. Y. Y. Chen, W. L. Chen and S. H. Huang, "Developing Arbitrage Strategy in High-frequency Pairs Trading with Filterbank CNN Algorithm," 2018 *IEEE International Conference on Agents (ICA)*, Singapore, 2018, pp. 113-116, doi: 10.1109/AGENTS.2018.8459920.
4. J. H. Liou, Y. T. Liu, and L. C. Cheng, "Price spread prediction in high-frequency pairs trading using deep learning architectures," *Int. Rev. Financ. Anal.*, vol. 96, p. 103793, 2024, doi: 10.1016/j.irfa.2024.103793.
5. X. Xu and Y. Zhang, "Neural network predictions of the high-frequency CSI300 first distant futures trading volume," *Financ. Mark. Portf. Manag.*, vol. 37, no. 2, pp. 191–207, 2023, doi: 10.1007/s11408-022-00421-y.
6. R. Chen and B. Pan, "Chinese stock index futures price fluctuation analysis and prediction based on complementary ensemble empirical mode decomposition," *Math. Probl. Eng.*, vol. 2016, no. 1, p. 3791504, 2016, doi: 10.1155/2016/3791504.
7. G. Li, X. Chen, and Y. Liu, "High-frequency lead-lag relationships in the Chinese stock index futures market: Tick-by-tick dynamics of calendar spreads," *arXiv preprint arXiv:2501.03171*, 2025, doi: 10.48550/arXiv.2501.03171.
8. Y. Zhao and D. Wan, "Institutional high-frequency trading and price discovery: Evidence from an emerging commodity futures market," *J. Futures Mark.*, vol. 38, no. 2, pp. 243–270, 2018, doi: 10.1002/fut.21888.
9. J. Hao, X. Song, F. He, *et al.*, "Price discovery in the Chinese stock index futures market," *Emerg. Mark. Finance Trade*, vol. 55, no. 13, pp. 2982–2996, 2019, doi: 10.1080/1540496X.2019.1598368.
10. Y. Hou and S. Li, "Volatility behaviour of stock index futures in China: A bivariate GARCH approach," *Stud. Econ. Finance*, vol. 32, no. 1, pp. 128–154, 2015, doi: 10.1108/SEF-10-2013-0158.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of GBP and/or the editor(s). GBP and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.