Structural break-aware pairs trading strategy using deep reinforcement learning

Jing-You Lu[1] · Hsu-Chao Lai[2] · Wen-Yueh Shih[2] · Yi-Feng Chen[2] ·
Shen-Hang Huang[1] · Hao-Han Chang[3] · Jun-Zhe Wang[4] · Jiun-Long Huang[1,2] ·
Tian-Shyr Dai[5]

Abstract

Pairs trading is an effective statistical arbitrage strategy considering the spread of paired stocks in a stable cointegration relationship. Nevertheless, rapid market changes may break the relationship (namely structural break), which further leads to tremendous loss in intraday trading. In this paper, we design a two-phase pairs trading strategy optimization framework, namely *structural break-aware pairs trading strategy* (*SAPT*), by leveraging machine learning techniques. Phase one is a hybrid model extracting frequency- and time-domain features to detect structural breaks. Phase two optimizes pairs trading strategy by sensing important risks, including structural breaks and market-closing risks, with a novel reinforcement learning model. In addition, the transaction cost is factored in a cost-aware objective to avoid significant reduction of profitability. Through large-scale experiments in real Taiwan stock market datasets, SAPT outperforms the state-of-the-art strategies by at least 456% and 934% in terms of profit and Sortino ratio, respectively.

Keywords Pairs trading strategy · Structural break detection · Deep reinforcement learning · Continuous wavelet CNN

✉ Jiun-Long Huang
jlhuang@cs.nctu.edu.tw

[1] Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

[2] Department of Computer Science, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

[3] Institute of Finance, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

[4] Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin, Taiwan

[5] Department of Information and Financial Management, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
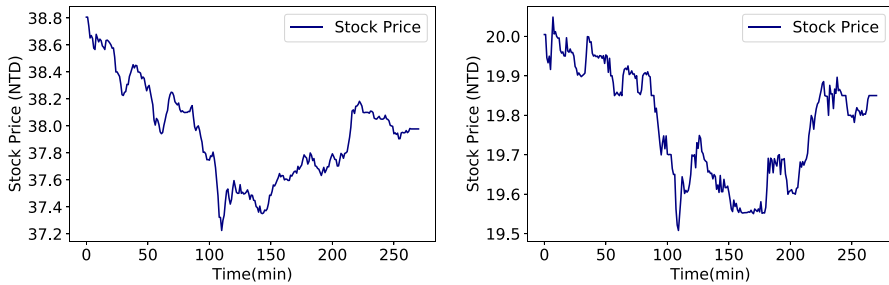
Fig. 1 Stock price of 2337.TW and 2344.TW

1 Introduction

With the advancement of communication techniques, information dissemination becomes faster than in the past, which means more external factors, such as breaking news, can impact the financial markets in real time so that it increases the volatility of financial markets and causes high investment risks. For investors, it becomes difficult to seize the opportunity for profit and consider the investment risk for stable return at the same time. To achieve such goals, the concept of pairs trading, a statistical arbitrage trading strategy, is proposed [35] and widely used in several financial markets [5, 37, 64].

Financial experts found that the spread of the prices of some pairs of financial instruments (i.e., the difference between their prices) is always in a stable long-term relation. Moreover, *pairs trading strategies* are developed to exploit these stable relations as the arbitrage opportunities (detailed later). Different definitions of meaningful pairs are then proposed, such as distance-based [27, 35], cointegration-based [14, 34], stochastic-based [56, 58], and time series-based [16, 21]. In this paper, we focus on the cointegration-based pairs trading[1] since the spread of a cointegration-based pair is proved to be more econometrically reliable [52, 66].

Figure 1 illustrates the stock prices of Macronix International Co., Ltd (2337. TW) and Winbond Electronics Corp. (2344.TW) in the Taiwan stock market on Jan 3, 2020, respectively. Both of them manufacture DRAM-related products, and the cointegration relationship of their prices is verified by [28]. The spread normalized by z-score is further shown in Fig. 2, where the gray area is the formation period to test whether they have cointegration relationship.

In the pairs trading scenario, a trade, composed of one *short* and one *long*, *opens* when an arbitrage opportunity occurs. Until the spread reverts to its historical mean, the trade is *closed* by doing the opposite actions. For example, in Fig. 2, the investors should short stock 2337.TW and long stock 2344.TW at the 177th minute, and close the position at the 191st minute by longing stock 2337.TW and shorting stock 2344.TW. Specifically, in pairs trading, there is a pair of trading boundaries (the green dashed lines), to triggers trades while the spread touches it. When the spread

---

[1] Pairs trading will stand for cointegration-based pairs trading in the following pages.
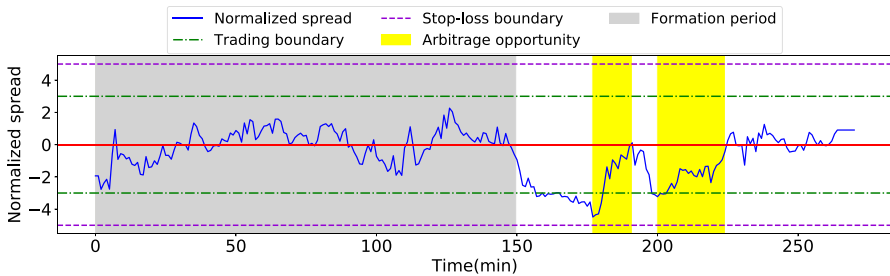
Fig. 2 Example of the normalized spread of paired stocks

returns to the historical mean (the red solid line), the trade would be recommended to *close* the position to make the profit. As a result, the yellow areas, bounded by these three lines, are the arbitrage opportunities.

However, the spread may diverge too far from historical mean unexpectedly. To avoid great loss caused by the divergence, the investors would set a pair of *stop-loss* boundaries (the purple dashed lines in Fig. 2), which is wider than the trading boundaries, to force the trade to close the position. On the other hand, if a trade is opened but the spread does not revert back to historical mean until a deadline, the trade would be forced to close the position as well, called *exit*. For example, in intraday trading, the deadline is set to the closing time of the day. In general, once stop-loss or exit happens, they usually result in a negative return [50]. Overall, the behavior of pairs trading is to offset the systematic risk by the positions of two different assets. Therefore, it has been regarded as a market neutral trading strategy with good hedging ability [35].

In order to optimize pairs trading strategies depending on the positions of open and close, it is crucial to determine the trading and stop-loss boundaries. If the gap between trading boundaries is narrow, the arbitrage is low, and hence, few profit can be made. Moreover, the little profit could be consumed by *transaction costs*, such as transaction taxes and fees. In contrast, if it is too wide, the strategies could not only miss several minor arbitrage opportunities but also increase the risk of great loss. For example, in Fig. 2, if the trading boundaries are set to ±2, the return is less than the boundaries of ±3. In contrast, although setting the boundaries to ±4 can create more profit in the first arbitrage opportunity, the second arbitrage opportunity is missed since the stock price did not meet the trading boundary. Overall, it is challenging but necessary to strike a good balance between the arbitrage opportunities and the risk control while learning the optimized trading and stop-loss boundaries.

To seize the arbitrage opportunities in intraday trading, our intuition is to design new pairs trading strategies in a fine-grained scale (e.g., minute scale) from the tick data. Note that most of the existing pairs trading strategies [35, 50, 66] were designed for daily data, in which the spreads are usually stable and long-term equilibrium. While the stock markets are easily influenced by some external factors (e.g., news and government policies) in real time [42], they may lose the arbitrage opportunities in intraday trading. On the other hand, the cointegration relationship of spreads is much weaker due to the high sensitivity in the tick data
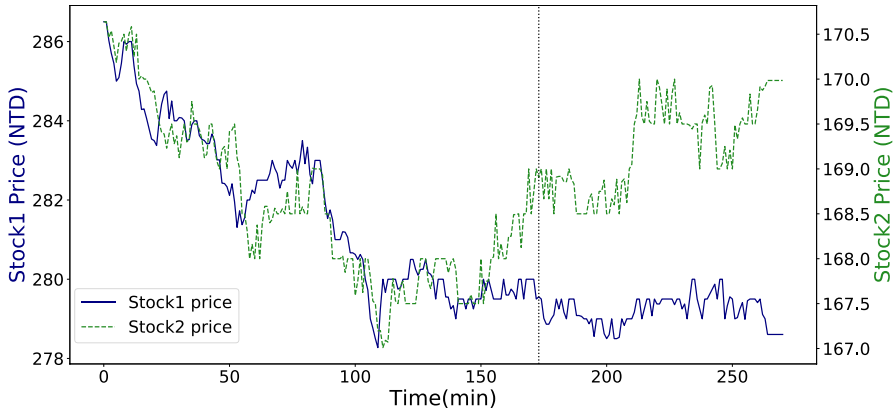
(a) Stock Prices of 1303.TW and 1319.TW
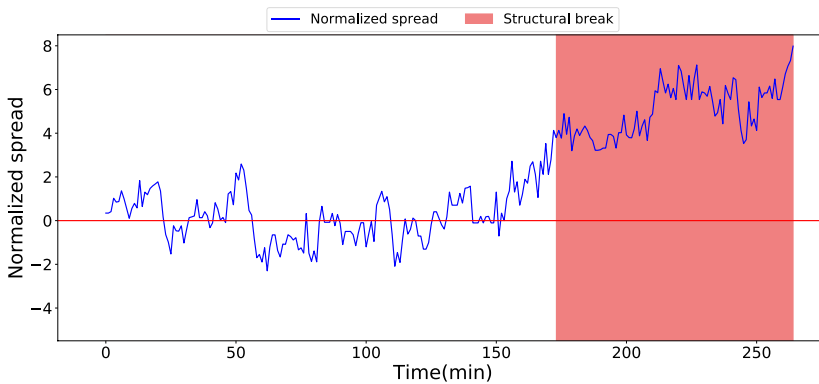
(b) Spread of 1303.TW and 1319.TW

Fig. 3 Example of the structural break

[42]. The risk, namely the *structural break*, that the spread could fade away from the historical mean also increases, which would cause dramatic loss if the spread does not revert back. The pink shaded area in Fig. 3 illustrates an example of a structural break. From the 173rd minute, which is a *breakpoint*, the spread of stocks 1303.TW and 1319.TW increases dramatically, and hence, the cointegration relationship vanishes. Pairs trading may suffer from a great loss if the strategy does not close the position when a breakpoint occurs. For instance, 16.2% of the trades executed by a state-of-the-art method PTDQN [50] in our Taiwan stock market dataset are forced to exit in the end of the trading period because PTDQN does not wisely close positions during structural breaks (detailed later). Therefore, the investments are put in risky conditions and 89.5% of these risky trades close with negative profit. Consequently, detecting structural breaks is important for pairs trading, but existing pairs trading strategies [35, 50, 66] seldom factor in the structural breaks to prevent huge loss.

To detect structural breaks, state-of-the-art methods, such as Augmented Dickey–Fuller test [25] and Chow test [20], require numerous data for statistical examinations. Moreover, they are not applicable to online detection. On the other hand, in anomaly detection field, likelihood ratio-based and probability-based change-point detection [1] identifies sudden and dramatic pattern changes in time series data. However, the structural breaks may lie in slowly changing spreads, which make them undetectable for change-point detection methods [4, 83]. To this end, we recognize there is an urgent need to design an effective structural break detection method to improve pairs trading strategy.

In this paper, we propose a two-phase framework, namely *structural break-aware pairs trading strategy* (*SAPT*), to tackle the above issues. The details of the two phases are listed below.

- **Phase 1: structural break detection**. Given a cointegrated pair of stocks, their previous stock price sequences, and their current stock prices, the goal is to estimate *the occurrence probability of a structural break* in the current timestamp. Inspired by recent works on time series data analysis [23, 65, 85, 87], combining the time-domain and the frequency-domain features can make significant improvement, compared to only considering either of them. It is worth noting that, besides time-domain features, frequency-domain information extracted by Fourier transform or wavelet transform has been proved to be effective for analyzing time series data [10, 65]. In particular, the finance data are usually of latent multi-frequency market patterns, such as seasonal behaviors, which can be extracted by analyzing the frequency-domain features [8, 85, 87]. Therefore, we propose s*pread* w*avelet-a*ware hybrid <u>net</u>*work* (*SWANet*) to jointly extract frequency-domain features from spreads and time-domain features with a continuous wavelet convolutional neural network (CNN) and a long short-term memory (LSTM), respectively. Through combining the two different aspects, SWANet is better handling nonlinearity and complexity in stock data than statistic approaches, such as auto-regressive integrated moving average model (ARIMA) [12]. Through SWANet, the predicted probability of structural break can be obtained and passed to the next phase as one of the risk features for further trading optimization.
- **Phase 2: pairs trading strategy optimization**. Given a cointegrated pair of stocks, their previous stock price sequences, the probability of structural break occurrence from phase 1, and the transaction cost defined by the markets, the goal of this phase is to jointly decide the trading shares (i.e., trading amount) in each timestamp, the trading and the stop-loss boundaries. In addition, we argue that risk control, including structural breaks and market-closing risks, is important in the noisy intraday trading scenario. Nonetheless, the previous literature [30, 70] fails to take them into consideration. As a result, we propose a novel deep Q-network s*tructural break-a*ware d*eep Q*-network (*SADQN*) with a transaction cost-aware objective function and risk-aware definitions of states and rewards. Consequently, SADQN incorporates not only profit but also the awareness of risks.

For evaluations, we conduct a large-scale dataset collected from the top 150 companies in the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX) from November 1st, 2017 to May 31st, 2020. The experimental results manifest that SWANet outperforms the state-of-the-art structural break detection methods 30.4% in terms of miss rate. For pairs trading strategy, compared to the state-of-the-art pairs trading strategy optimization methods, SADQN increases 456% of profit and 934% of Sortino ratio in the sense of risk control, respectively. The main contributions of this paper are listed as follows:

1. To the best of our knowledge, in machine learning domain, we are the first to identify the urgent need of developing pairs trading strategy with structural break detection in intraday trading scenario.
2. We propose a novel structural break detection method SWANet, which considers both frequency-domain and time-domain features, to detect the breakpoints of cointegrated pairs efficiently.
3. We design a new deep Q-network SADQN which factors in the structural breaks, market-closing risks, and the transaction costs to optimize the pairs trading strategy.
4. We collect a large-scale tick data from the Taiwan stock market for experiments. The results manifest our solutions outperform the state-of-the-art methods significantly.

The rest of this paper is organized as follows. The related works are compared in Sect. 2. We overview the background of pairs trading strategy and the proposed framework SAPT in Sect. 3. The details of our models SWANet and SADQN are, respectively, presented in Sects. 4 and 5. Finally, Sect. 6 shows the experimental results and Sect. 7 concludes this paper.

2 Related work

2.1 Structural break detection

2.1.1 Statistical methods for structural break detection

Statistical methods [20, 25] are generally used to test the stationarity of a time series. Augmented Dickey–Fuller test (ADF) [25] borrowed the idea "unit root," which describes the condition that a random walk of a time series cannot be fit by a linear statistical model [88], to identify non-stationary data and then reported it as a structural break. Chow test [20] examined whether the coefficients in two linear regressions of two subsequences, which are, respectively, extracted from a time series before and after a certain point, are equal and reported as a structural break if not. These methods are too sensitive to deal with the high variety in intraday trading. In addition, they have difficulty to meet the immediacy of structural break detection since they require enormous data to get reliable results during testing.

2.1.2 Change-point detection

The goal of change-point detection is to identify the locations where pattern switches or abrupt change happens in the time series data. It has been applied to a broad range of real-world application domains [3, 18, 83], e.g., climate change detection and speech recognition. Adams et al. [1] estimated the Bayesian probability of how possible a data point turns into a change-point by referring to the probability of recent data points. Kang et al. [46] first extracted features from time series by Fourier transform and then exploited vector quantization clustering to detect change points which switch clusters over time. Nevertheless, the above methods may misclassify abrupt fluctuations in stock markets as breakpoints, such that many arbitrage opportunities would be missed.

2.2 Pairs trading strategy

2.2.1 Pairs trading

Pairs trading is a market neutral trading strategy that has been widely used since the 1980s [35]. By constructing a pair of two stocks, pairs trading can offset the systematic risk, especially in the volatile market, and make the profit with excellent hedging ability. Gatev et al. [35] proved that the average annualized excess return by the US daily stock data from 1962 to 2002 can be promoted to 11% by employing simple pairs trading strategy. There are several types of approaches for pairs trading [52]. For example, Gatev et al. [35] and Do and Faff [27] utilized the distance metrics to find the pair of stocks whose prices move together. Some studies [14, 34] found the pairs with cointegration relationship, and some [56, 58] modeled the spread of paired stock prices by Ornstein–Uhlenbeck process. Since some studies have shown that the cointegration method has generated a more stable and robust excess return [52, 66], in this paper, we find the paired stocks by verifying the cointegration relation between two stocks for further trading. However, as most of the studies [14, 34] above focus on the identification of paired stocks and test the result with basic methods, such as static standard deviation as open and stop-loss boundaries. In this paper, we focus on how to decide the open and stop-loss boundaries dynamically to trigger trades in pairs trading.

2.2.2 Deep learning for finance

Due to the complexity of financial markets, the nonlinear characteristics of stock price may not meet the statistical assumptions [88]. To this end, deep neural networks have been exploited to forecast stock prices or detect outliers nowadays [24, 82, 85, 87]. Chen et al. [19] constructed the Filterbank CNN in high-frequency pairs trading to extract long-term and short-term historical volatility information and hence outperformed the rule-based strategies in Taiwan Stock Index Futures and Mini Index Futures. Zhang et al. [86] proposed to extract the features of limit order books with a CNN to learn the strength of the bid–ask and then predict the trend

of the London Stock Exchange with an LSTM. Several studies [8, 24] improved the robustness by utilizing neural networks to alleviate the effect of noise and uncertainty.

Specifically, reinforcement learning models have shown great performance in optimizing trading decisions in the financial domain [13, 24, 30, 50]. For instance, many financial trading applications based on reinforcement learning focus on making trading decisions of a single stock at each timestamp [24, 55, 57]. For pairs trading, Fallahpout et al. [30] made the first attempt to adopt a classic Q-learning model. Kim et al. [50] further proposed the pairs trading deep Q-network (PTDQN) that employed a conventional deep Q-learning model to dynamically choose the open and the stop-loss boundaries over time. The above methods are designed for daily data and they hold the shares for a long period in which the spreads they find are in a stable and long-term equilibrium. Nonetheless, in practice, the stationary of paired stocks is oscillating in intraday trading, especially when the close time of the market is approaching. Note that it is more profitable, but also risky, in the oscillating intraday trading environment. Moreover, those methods did not count in the transaction cost, which may easily decrease the profit, during training. To strike a good balance between profitability, risks, and transaction cost in intraday trading scenario, we propose a novel reinforcement learning model SADQN, where the structural breaks, market-closing risk, and transaction cost are all factored in.

3 Overview

We first briefly introduce the background knowledge of cointegration relationship in Sect. 3.1. An overview of the proposed framework structural break-aware pairs trading strategy (SAPT) in further presented in Sect. 3.2.

For clarity of presentation, in this paper, non-bold lowercase letters (e.g., x) and non-bold uppercase letters (e.g., X) denote scalars and sets, respectively. Bold uppercase letters (e.g., $\mathbf{X}$) and bold lowercase letters (e.g., $\mathbf{x}$) denote matrices and vectors, respectively.

3.1 Stationary and cointegration relationship

In finance [6, 49, 77] and time series studies [36, 38, 62], a time series is "stationary" if (1) the expectation of the series over time is a constant, (2) the variance of the series over time is a constant, and (3) the auto-covariance of the series of two timestamps only depends on a lagged value. In other words, a time series is stationary if it is very stable along time. Note that, in stock markets, non-stationary stocks are more profitable than stationary stocks since the prices of the former could rise more potentially.

However, non-stationary stocks have to take greater risks of price tumble. In order to cope with both the profitability and the risk control, financial experts propose the pairs trading strategy based on the cointegration relationship [29], which linearly combines two non-stationary stocks into one stationary time series. Among
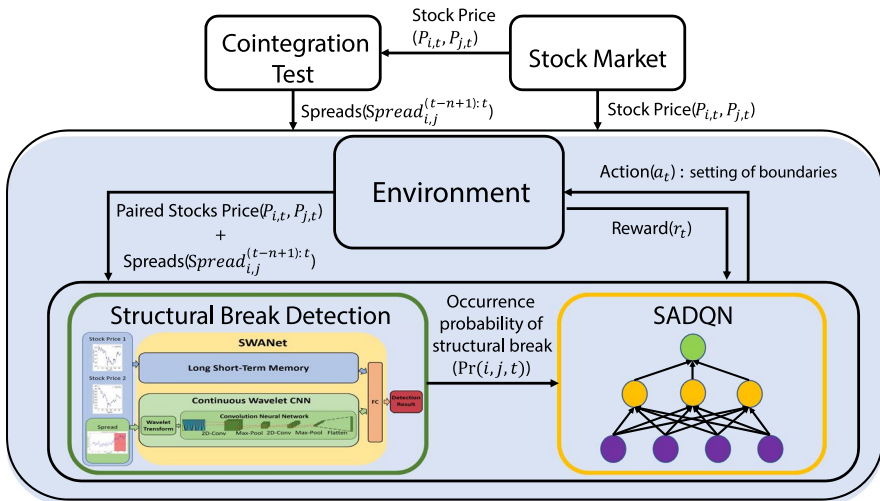
Fig. 4 Architecture of SAPT

the literature of cointegration relationships [29], in this paper, we adopt the vector error correction model (VECM) [14, 30] to extract cointegrated pairs of stocks with statistical tests. Finally, the spread of a cointegrated pair composed of $Stock_i$ and $Stock_j$ at timestamp t, termed as $Spread_{i,j}^t$, is formulated as follows:

$$Spread_{i,j}^t = h_i \cdot p_{i,t} + h_j \cdot p_{j,t}, \tag{1}$$

where $p_{i,t}$ and $p_{j,t}$ denote the stock prices of $Stock_i$ and $Stock_j$ at t. h_i and h_j, which are determined by VECM, respectively, weights $Stock_i$ and $Stock_j$. As a result, $Spread_{i,j}^t$ is the difference of weighted stock prices between the $Stock_i$ and $Stock_j$. It is worth noting that, in pairs trading strategy, the ratio between trading amounts of $Stock_i$ and $Stock_j$ is regulated to $\frac{h_i}{h_j}$ to keep the cointegration relationships after a trade is made.

3.2 Overview of SAPT

To facilitate structural break-aware pairs trading strategy, two key tasks are to: (1) detect structural breaks in trading periods and (2) determine the trading policy with risk information and transaction cost. The blue shaded area in Fig. 4 illustrates the architecture of the proposed two-phase machine learning framework structural break-aware pairs trading strategy (SAPT). The first-phase SWANet (the green rectangle) estimates the occurrence probability of structural breaks with a hybrid model learning from time- and frequency-domain features. The second-phase SADQN (the yellow rectangle) determines the setting of boundaries (including trading and stop-loss boundaries) over time dynamically with a novel reinforcement learning model, where a transaction cost-aware objective and a risk-aware environment setting

are incorporated. The details of SWANet and SADQN are formally introduced in Sects. 4 and 5, respectively.

4 Structural break detection

4.1 Problem definition

Pairs trading arbitrages when a stationary spread reverts to the historical mean. Nevertheless, the stationary spread may turn into a non-stationary time series due to external impact (e.g., news or government policies). This event is called a *structural break* [39], and the specific timestamp when the structural break occurs is named a *breakpoint*. Although the occurrence of structural breaks may result in great profit in pairs trading if the spread returned to the historical mean eventually, it could also cause a great loss if the spread does not revert to the historical mean at all. To prevent such high-risk situations, it is important to detect structural breaks in pairs trading. Consider a cointegrated pair of stocks $Pair_{i,j} = \langle Stock_i, Stock_j \rangle$, where $Stock_n = \langle p_{n,t} \in R^+ \mid t = 1, 2, \ldots, t^{cur} - 1 \rangle$ for all n, and let t^{cur} denote the current timestamp. When the current prices $p_{i,t^{cur}}$ and $p_{j,t^{cur}}$ are available, the goal is to estimate the occurrence probability of a structural break at t^{cur} with a real-time detection model f_θ as follows:

$$Pr(i, j, t^{cur}) = f_\theta(Pair_{i,j}, t^{cur}), \tag{2}$$

where Pr is the occurrence probability, θ is the set of learnable parameters of f_θ. Accordingly, we further model the detection task as a binary classification problem, where the objective function is defined as follows:

$$\mathcal{L} = \sum_{\forall Pair_{i,j} \in DB} \sum_{t=t^{start}}^{T} y_{i,j,t} \log Pr(i, j, t) + (1 - y_{i,j,t}) \log(1 - Pr(i, j, t)),$$

$$\text{where } y_{i,j,t} = \begin{cases} 1, & \text{structural break occurs} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

DB is a database collecting all the cointegrated pairs. t^{start} denotes the starting timestamp which is available to trade (i.e., the very next timestamp after the formation period) and T denotes the last timestamp of the day. $y_{i,j,t}$ is the binary ground truth of whether $Pair_{i,j}$ is in a structural break at time t. To be more specific, $y_{i,j,t} = 1$ indicates that the spread at time t is under structural break, and $y_{i,j,t} = 0$ otherwise. For example, if $t \in \{1, 2, 3, 4, 5\}$ and structural break happens at the third and fourth timestamp, y is $\{0, 0, 1, 1, 0\}$. The ground truth of breakpoint can be obtained by [54]. Consequently, the objective is to minimize the binary cross-entropy $\mathcal{L}$ of all cointegrated pairs in DB in the trading period (i.e., $t \in \{t^{start}, \ldots, T\}$).

The differences between the proposed SWANet and traditional statistical methods [20, 25] are twofold. (1) Statistical methods require relatively long time for detection and are designed for offline examinations. SWANet is able to detect structural breaks online and hence reduces the detection delay. (2) SWANet learns from
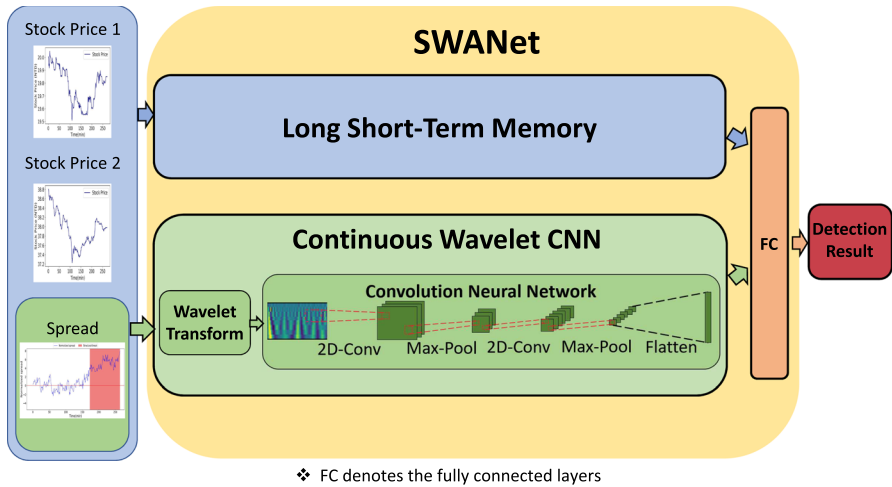
Fig. 5 Architecture of SWANet

both time-domain and frequency-domain features, whereas traditional methods do not exploit frequency-domain features. As illustrated in Fig. 5, SWANet combines a CNN with continuous wavelet transform and an LSTM to extract the frequency-domain and time-domain features, respectively. Then, the extracted features are forwarded to the fully connected layer (FC in Fig. 5), and the detection result is the predicted probability of structural break. We detail the continuous wavelet CNN and the combined model in Sects. 4.2 and 4.3, respectively.

4.2 Continuous wavelet CNN

In signal processing, Fourier transform [69, 78] is useful to derive frequency-domain representations of regular time series, such as electromagnetic signals [2, 79] and factory machine signals [7, 31]. However, the time series of stock prices could vary dramatically over time, which is not suitable to use Fourier transform. In contrast, continuous wavelet transform (CWT) [68, 81], which derives frequency-domain representations of a signal by continuously changing scaling and shifting some wavelets, is proved to be effective in handling such time-varying signals.

Specifically, CWT changes the scale and the location of a mother wavelet $\psi_{a,b}(t)$ to represent the target signal, where $a \in R^+$ and $b \in R$ denote *scale* and *shift*, respectively. Accordingly, CWT is defined as follows:

$$\mathrm{CWT}_x(a,b) = \frac{1}{\sqrt{a}} \int x(t)\psi\left(\frac{t-b}{a}\right)\mathrm{d}t, \tag{4}$$

where x is the target signal. CWT can be regarded as the similarity between the target signal $x(t)$ and mother wavelet $\psi_{a,b}(t)$. In this paper, we adopt *Ricker wavelet*
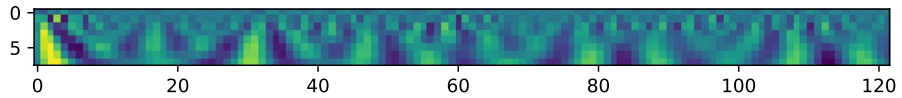
Fig. 6 An example of scalogram

as the mother wavelet since the performance is better. After conducting CWT, the results of the wavelet coefficients can be assemble into a scalogram (detailed later).

Note that the scalogram contains rich frequency-domain information. As a result, our idea is to exploit the CNN model to extract frequency-domain features from *the scalogram of spreads* to detect structural breaks. Given a cointegrated pair $Pair_{i,j}$ as an input, we first derive its spread $\text{Spread}_{i,j}$ in the trading period as follows:

$$\text{Spread}_{i,j} = \langle \text{Spread}_{i,j}^{t^{\text{start}}}, \text{Spread}_{i,j}^{t^{\text{start}}+1}, \ldots, \text{Spread}_{i,j}^{T}, \rangle \tag{5}$$

where each $\text{Spread}_{i,j}^{t}$ follows the definition in Eq. (1), and hence, $\text{Spread}_{i,j}$ is the sequence consisting of the spread of $Pair_{i,j}$ of each timestamp in the trading period. The target signal of CWT is further set as $\text{Spread}_{i,j}$, i.e., $CWT_{\text{Spread}_{i,j}}(a,b)$, to obtain the scalogram of $Pair_{i,j}$, termed as $SG_{i,j}$. Figure 6 illustrates an example of a scalogram $SG_{i,j}$, where the *X*- and *Y*-axes denote the timestamp and the wavelet frequency of spread, respectively. A pixel at coordinate (*x*, *y*) is yellow represents that frequency *y* is strong at timestamp *x*, and blue otherwise. Through this process, we expect that some special oscillations (e.g., abrupt chasms and gradual changes) could be extracted from the scalogram.

CNN has been proved to be effective in extracting significant features from images [59, 67, 84], time series [19, 44, 60], etc. However, many applications in time series analysis apply a one-dimensional CNN on the univariate time series. We argue that such method does not make the best use of the information in data and the robustness to deal with multi-dimensional data in CNN. We propose to use a two-dimensional CNN instead, because we expect that the nonlinear convolution can naturally capture the interplay of features across the two dimensions in the scalogram.

Given a spread scalogram $SG_{i,j}$, we first generate a feature set $X_{i,j}^{CNN}$ with a predefined window size *d* in order to apply to the online detection system. It is defined as follows:

$$X_{i,j}^{CNN} = \{\mathbf{X}_{i,j,t}^{CNN} | \mathbf{X}_{i,j,t}^{CNN} = SG_{i,j}(t-d:t-1,:), t = t^{\text{start}}, t^{\text{start}}+1, \ldots, T\}, \tag{6}$$

where $SG_{i,j}(t-d:t-1,:)$ is the partial scalogram that the *x*-axis is collected from timestamp $t-d$ to $t-1$, and is the raw frequency-domain feature of $Pair_{i,j}$ at time *t*. Note that $X_{i,j}^{CNN}$ only collects data for detecting breakpoints in trading periods. The two-dimensional convolution layers are then defined as follows:

$$\mathbf{M}^{(1)} = \phi(\mathbf{W}_{\text{conv}}^{(1)} \otimes \mathbf{X}_{i,j,t}^{CNN} + \mathbf{b}_{\text{conv}}^{(1)}) \tag{7}$$

$$\mathbf{M}^{(l)} = \phi(\mathbf{W}_{\text{conv}}^{(l)} \otimes \mathbf{M}^{(l-1)} + \mathbf{b}_{\text{conv}}^{(l)}), \tag{8}$$

where l is the numbering of a layer. $\otimes$ is the convolution operation with a learnable kernel $\mathbf{W}_{\text{conv}}^{(l)}$. $\mathbf{b}_{\text{conv}}^{(l)}$ is the learnable bias to stabilize the learning process and ϕ is an activation function, which is set to ReLU in this paper. The input of the continuous wavelet CNN is the partial scalogram $\mathbf{X}_{i,j,t}^{CNN}$, as shown in Eq. (7). Compared to using one-dimensional layer for time series [73, 75], the input and the output of Eq. (8) are two-dimensional matrices (i.e., $\mathbf{M}^{(l-1)}$ and $\mathbf{M}^{(l)}$), which are able to extract more complicated signals from the two-dimensional scalograms. As shown in Fig. 5, the convolutional layers are stacked and every convolution layer is followed by a max pooling layer to avoid over-fitting and to reduce computational effort.

4.3 Integrating continuous wavelet CNN and LSTM

After introducing the continuous wavelet CNN for extracting frequency-domain features, we further describe how SWANet exploits an LSTM model to extract time-domain features and then how they are integrated for structural break detection.

LSTM is one of the most widely used recurrent structures in sequence modeling, and has been widely deployed in natural language processing [76, 80], video captioning [26, 33], time series analysis [15, 47, 74], etc. The success of LSTM may credit to its great capability of memorizing long-term memories while identifying which memory should be forgotten. To detect structural breaks for pairs trading, we utilize LSTM to recognize the fluctuation patterns in the spreads of cointegration pairs. Unlike traditional methods that analyze univariant time series, we input not only the spread ($Spread_{i,j}$) but also the price values of two stocks ($Stock_i$ and $Stock_j$) to LSTM, as illustrated in Fig. 5. Therefore, when the paired stocks move together, the spread may remain stationary while the fluctuations of the stock prices can be sensed. Specifically, for each timestamp t, the input is $\mathbf{x}_{i,j,t}^{LSTM} = [Spread_{i,j}^t, \hat{p}_{i,t}, \hat{p}_{j,t}]^\top$, where $\hat{p}_t$ stands for the normalized stock price, and X^{LSTM} is the set collecting all $\mathbf{x}_{i,j,t}^{LSTM}$. Please note that since the range of price varies from one stock to another and the structural break is related to the changes of price, the normalized price is more suitable for structural break detection than raw stock price. Furthermore, the components of LSTM are defined as follows:

$$\mathbf{i}_{i,j,t} = \sigma(\mathbf{W}_{\text{input}} \mathbf{x}_{i,j,t}^{LSTM} + \mathbf{U}_{\text{input}} \mathbf{h}_{i,j,t-1} + \mathbf{b}_{\text{input}}) \tag{9}$$

$$\mathbf{f}_{i,j,t} = \sigma(\mathbf{W}_{\text{forget}} \mathbf{x}_{i,j,t}^{LSTM} + \mathbf{U}_{\text{forget}} \mathbf{h}_{i,j,t-1} + \mathbf{b}_{\text{forget}}) \tag{10}$$

$$\mathbf{o}_{i,j,t} = \sigma(\mathbf{W}_{\text{output}} \mathbf{x}_{i,j,t}^{LSTM} + \mathbf{U}_{\text{output}} \mathbf{h}_{i,j,t-1} + \mathbf{b}_{\text{output}}) \tag{11}$$

$$\mathbf{c}_{i,j,t} = \mathbf{i}_{i,j,t} \circ \tilde{\mathbf{c}}_{i,j,t} + \mathbf{f}_{i,j,t} \circ \mathbf{c}_{i,j,t-1} \tag{12}$$

$$\tilde{\mathbf{c}}_{i,j,t} = \tanh(\mathbf{W}_{\text{cell}}\mathbf{x}_{i,j,t}^{LSTM} + \mathbf{U}_{\text{cell}}\mathbf{h}_{i,j,t-1} + \mathbf{b}_{\text{cell}}) \qquad (13)$$

$$\mathbf{h}_{i,j,t} = \mathbf{o}_t \circ \tanh(\mathbf{c}_{i,j,t}), \qquad (14)$$

where $\mathbf{W}_*$ and $\mathbf{U}_*$ are learnable weight matrices, and $\mathbf{b}_*$ are learnable bias vectors. Equations (9), (10), and (11) formulate the input gate, forget gate, and output gate, respectively. $\mathbf{c}_{i,j,t}$ is the data stored in the cell, and $\tilde{\mathbf{c}}_{i,j,t}$ is the candidate of the cell. σ denotes the sigmoid activation function and $\circ$ denotes the Hadamard product (i.e., element-wise product). The new value in cell at time t is aggregated from the current candidate $\tilde{\mathbf{c}}_{i,j,t}$ and the value already stored in the cell $\mathbf{c}_{i,j,t}$ filtered by the input gate and the forget gate, respectively. $\mathbf{h}_{i,j,t}$ is both the hidden state at time t and also the output of LSTM. During training, for each $Pair_{i,j}$, we input the corresponding $\mathbf{x}_{i,j,t}^{LSTM} \in X^{LSTM}$ sequentially based on the order of time.

By conducting the continuous wavelet CNN and the LSTM, SWANet extracts frequency-domain and time-domain features of cointegrated pairs. To integrate and learn the interaction of those extracted features, the solution is to stack several fully connected layers, where the input is the concatenation of the two features. Note that the concatenation is adopted, rather than sum, since it is possible to learn nonlinear interactions between each element via those layers. The fully collected layers $\mathbf{l}_{\text{full}}^{(*)}$ are defined as follows:

$$\mathbf{l}_{\text{full}}^{(1)} = \sigma(\mathbf{W}_{\text{full}}^{(1)}(\mathbf{m}_{i,j,t} \oplus \mathbf{h}_{i,j,t}) + \mathbf{b}_{\text{full}}^{(1)}) \qquad (15)$$

$$\mathbf{l}_{\text{full}}^{(l)} = \sigma(\mathbf{W}_{\text{full}}^{(l)}\mathbf{l}_{\text{full}}^{(l-1)} + \mathbf{b}_{\text{full}}^{(l)}), \qquad (16)$$

where $\mathbf{W}_{\text{full}}^{(*)}$ and $\mathbf{b}_{\text{full}}^{(*)}$ are the learnable weight matrices and bias vectors in fully connected layers. For a input $Pair_{i,j}$ at time t, $\mathbf{m}_{i,j,t}$ denotes the vector of frequency-domain features, which is the flatten output of the continuous wavelet CNN (i.e., $\mathbf{M}^{(L)}$). Note that the frequency-domain features $\mathbf{m}_{i,j,t}$ and time-domain features $\mathbf{h}_{i,j,t}$ are the input of the stacked layers, where $\oplus$ denotes the concatenation. The output $\mathbf{l}_{\text{full}}^{(L')}$ of the last layer L' is a scalar, which is the occurrence probability of a structural break $Pr(i, j, t)$. The model is trained end to end by minimizing the cross-entropy formulated in Eq. (3). In addition, we adopt dropout to relieve over-fitting.

5 Pairs trading strategy optimization

5.1 Problem definition

In pairs trading, the perfect arbitrage opportunities occur between: (1) a diverging spread starts to revert to its historical mean (i.e., a perfect timing to *open position*) and (2) the spread meets the historical mean (i.e., a perfect timing to *close position*). It is crucial to decide the open and close positions to make profit for pairs trading strategies. Let $\text{Spread}_{i,j}$ denote the spread of a cointegrated pair of stocks i and j, and $Stock_i$ and $Stock_j$ be their respective stock price sequences.

$q_{i,j,t} \in \{\text{OPENED}, \text{CLOSED}\}$ denotes the trading status of the pair at time t. $q_{i,j,t} = \text{OPENED}$ if the position has been opened earlier and is waiting for a close; $q_{i,j,t} = \text{CLOSED}$ if the position has not been opened yet and is waiting for an open. Note that the status is set to CLOSED by default when the trading period starts every day. Following [50], the transaction cost (e.g., transaction tax or fee) is formulated as follow:

$$TC(v_{i,t}, v_{j,t}, t) = \mathcal{C} \times (|v_{i,t}| \times p_{i,t} + |v_{j,t}| \times p_{j,t}), \tag{17}$$

where $p_{i,t}$ and $v_{i,t} \in Z$, for stock i at time t, stand for the stock price and the volume of the trade in terms of shares, respectively. $\mathcal{C} \in [0, 1]$ is the transaction cost rate specified by each stock market.[2] Note that $v_{i,t}$ is positive for *longing* and negative for *shorting*, respectively. As a result, the transaction cost *TC* is the sum of stock prices weighted by the absolute trading volumes, and discounted by the cost rate. Note that, as mentioned in Sect. 3.1, $\frac{v_{i,t}}{v_{j,t}}$ should be equal to $\frac{h_i}{h_j}$ to maintain the cointegrated relationship.

Indeed, for pairs trading strategy optimization, the goal of this paper is to maximize the profit by deciding: (1) when to open, (2) when to close, and (3) the trading volume for all cointegrated pairs during the trading period. Accordingly, the *transaction cost-aware* objective function is defined as follows:

$$\arg\max_{V} \sum_{\forall Pair_{i,j} \in DB} \sum_{t=t^{\text{start}}}^{T} Profit(v_{i,t}, v_{j,t}, t), \tag{18}$$

where $V = \{(v_{i,t}, v_{j,t}, t)| \forall Pair_{i,j} \text{ and } t \in \{t^{\text{start}}, \dots, T\}\}$ is the set of trading volume and timestamps in trading periods. $Profit(v_{i,t}, v_{j,t}, t)$, denoting the profit earned at time t, is defined as follows:

$$
\begin{aligned}
&Profit(v_{i,t}, v_{j,t}, t) \\
&= \begin{cases} ((v_{i,t} \cdot (p_{i,t} - p_{i,t_{i,j}^{\text{open}}}) + v_{j,t} \cdot (p_{j,t} - p_{j,t_{i,j}^{\text{open}}})) - TC(v_{i,t}, v_{j,t}, t), & \text{if } q_{i,j,t} = \text{OPENED} \\ 0, & \text{if } q_{i,j,t} = \text{CLOSED} \end{cases}
\end{aligned} \tag{19}
$$

where $t_{i,j}^{\text{open}}$ is the timestamp when this pair is opened. That is, the profit gained from a pair of open and close positions is the sum of the stock price differences weighted by the trading volume, and then minus the transaction cost. Note that the profit can only be made when the cointegrated pair has been opened earlier and is ready to be closed.

In order to decide the open and close positions, finance experts proposed to set the trading boundaries and stop-loss boundaries with statistical methods [17, 32, 43]. Following their ideas, we model the pairs trading strategy as a Markov decision process (MDP) [9] and propose a deep reinforcement learning model

---

[2] For instance, $\mathcal{C} = 0.15\%$ for day trading securities transaction tax in Taiwan and $\mathcal{C} = 0.00221\%$ for Section 31 Transaction Fees in America stock markets, respectively.

SADQN to optimize the strategy by deciding the optimal boundaries. Contrary to those statistical methods considering stock prices and spreads only, SADQN covers more important aspects, including risk controls (detailed later).

5.2 Structural break-aware deep Q-network

Indeed, it is crucial to foresee the status of stocks since the market is dynamic. Model-free-based reinforcement learning models have shown significant usage in financial domains for stock price prediction [24, 63] as they can update their policy by sensing the market environment and evaluate their actions for future decisions. However, few attention on pairs trading strategy optimization has been drawn. In this paper, we propose SADQN, which adopts a deep Q-network (DQN) to build a Q-function based on historical events and estimates the Q-values with given states and actions. In the inference stage, the action with the highest Q-value is selected to maximize the potential profit. SADQN incorporates (1) risk factors in the states to avoid major risks in intraday pairs trading (e.g., structural breaks) and (2) transaction cost-aware rewards to maximize the total net profit. In this section, we model the pairs trading process as a Markov decision process (MDP) formulated as (S, A, T, R), which, respectively, denotes the tuple of a state set, an action set, transition probability, and a reward function, as follows.

- **State** $s^t \in S$: S is the set collecting all possible states in the environment. The state s^t generated from a cointegrated pair of stocks i and j at time t can be formulated as:

$$s^t = (\text{Spread}_{i,j}^{(t-n+1):t}, a^{t-1}, Pos_{i,j}^{t-1}, Pr(i,j,t), r_t) \tag{20}$$

  The elements of s^t are defined as follows:
- **The occurrence probability of structural break** $Pr(i, j, t)$: Without taking structural breaks into considerations, the loss could be great when that happens [48]. However, the previous literature does not factor in these important risk. As a result, $Pr(i, j, t)$ estimated by SWANet is included. We expect that SADQN will adjust the stop-loss boundary to close positions immediately if $Pr(i, j, t)$ increases.
- **Market-closing risk** (r_t): In intraday trading, the cointegrated pairs are forced to be closed at the end of the trading period if they have not been closed yet. However, this type of close, namely *exit*, usually results in less profit or even negative profit because the ideal close positions (i.e., historical mean) are not met. Without considering the approaching deadline of trading, more than 16% of trades executed by PTDQN [50] end up as exits and nearly 90% of those exits loses money. To avoid such loss, we define the market-closing risk as the ratio of the remaining time to the end of trading period $r_t = \frac{T-t}{T-t^{\text{start}}}$, where $t \in \{t^{\text{start}}, \ldots, T\}$. Therefore, r_t decreases when the deadline approaches, and keeps SADQN noticing the deadline.

Table 1 An example of action set A

	a_0	a_1	a_2	a_3	a_4	a_5	a_6
Trading boundary	± 0.5	± 1.0	± 1.5	± 2.0	± 2.5	± 3.0	$\pm \infty$
Stop-loss boundary	± 2.5	± 3.0	± 3.5	± 4.0	± 4.5	± 5.0	$\pm \infty$

- **Spread** $\text{Spread}_{i,j}^{(t-n+1):t}$: $\text{Spread}_{i,j}^t$ denotes the spread of stock i and j at time t and $\text{Spread}_{i,j}^{(t-n+1):t}$ is the sequence of spread from time $t - n + 1$ to t. That is,

$$\text{Spread}_{i,j}^{(t-n+1):t} = \langle \text{Spread}_{i,j}^{(t-n+1)}, \ \text{Spread}_{i,j}^{(t-n+2)}, \ \dots, \text{Spread}_{i,j}^t \rangle. \tag{21}$$

  Only the latest n timestamps are tracked in order to capture the up-to-date trends. Note that state s^t is independent to the identity of stocks but only considers the values of spread sequences. It is because two different cointegrated pairs with identical spread sequences are believed to have similar trends.
- **Previous action** a^{t-1}: Since each trade generates corresponding transaction cost, frequently open and close the trades may result in great accumulated transaction cost. To prevent from such condition, the action in the previous time t, termed as a^{t-1}, is included in the state by following a similar idea in [24]. The definition of action will be detailed later.
- **Previous position** $Pos_{i,j}^{t-1}$: Similar to the previous action above, changing positions (where a trade is opened) frequently may lead to heavy transaction cost [24]. As a result, we also factor in the previous position $Pos_{i,j}^{t-1} \in \{-1, 0, 1\}$ of the paired stock i and j in the state. To be more specific, at time $t - 1$, $Pos_{i,j}^{t-1} = 1$ indicates that the position is opened by meeting the upper trading boundary; $Pos_{i,j}^{t-1} = -1$ indicates that it is opened by meeting the lower trading boundary; otherwise, the position is not opened.

- **Action** $a \in A$: In practice, pairs trading strategies manipulate the trading boundaries and stop-loss boundaries as their policies. As a result, an action consists of those boundaries, whose unit is the standard deviation with regard to each spread. For example, Table 1 illustrates an action set $A = \{a_0, a_1, \dots, a_6\}$. The upper and the lower trading boundaries of a_0 are +0.5 and -0.5 standard deviations, respectively. Note that the action set is predefined and stored in a one-hot encoding representation in order to fit the input format of neural networks. In the case of Table 1, a_0 and a_5 are encoded as [1, 0, 0, 0, 0, 0, 0] and [0, 0, 0, 0, 0, 1, 0], respectively. Note that, in each action, there is a pair of symmetric trading boundaries. The gap between the two trading boundaries is regarded as the safe place for arbitrage. Similarly, there are two symmetric stop-loss boundaries with a wider gap. In contrast, if the spread goes beyond either stop-loss boundary, the spread may diverge and leads to great loss in pairs trading (details later). While the space between the boundaries is narrow, the position is opened and closed frequently, leading to lower risk as well as lower profit. In contrast, the trades are seldom triggered with wider spaces and hence have greater potential of

profitability. However, the risk of great loss increases as well. To prevent open/close unprofitable position, such as the states with high occurrence probability of structural break or high market-closing risks, we propose a *hold* action by setting both trading boundary and stop-loss boundary as $\pm\infty$ to prevent trading from loss, as a_6 in Table 1. There are two major advantages of incorporating the hold action: (1) when a trade has great potential to revert to historical mean quickly, it is not necessary to close the trade even the stop-loss boundary is met. By using the hold action, it is possible for SADQN to suspend the close and hence benefit from those more profitable arbitrage opportunities. (2) When spreads diverge too far from their historical means, even the widest boundary setting may not stop from opening those risky positions. However, by using the hold action, SADQN is able to forbid those opens to avoid such risks. While none of the previous methods [24, 30, 50] considers the hold action, we make the first attempt to adopt it in SADQN for the aforementioned advantages.

- **Transition Probability,** $T = Pr(s^{t+1}|s^t, a)$: The transition probability $T = Pr(s^{t+1}|s^t, a)$ estimates the probability of state s^{t+1} at time $t + 1$ condition to the given state s^t and the action a. Unlike model-based reinforcement learning that all the transition probabilities are given, SADQN learns them through data.
- **Reward Function,** $R(s^t, a^t, s^{t+1})$: In order to maximize the transaction cost-aware objective function formulated in Eq. (3), SADQN adopts a widely used concept, *net return* [50], in the reward function. The net return of a cointegrated pair of stocks i and j at time t is defined as follows:

$$NR_{i,j}^t = \left(v_{i,t} \times \frac{p_{i,t} - p_{i,t^{\text{open}}}}{p_{i,t^{\text{open}}}} + v_{j,t} \times \frac{p_{j,t} - p_{j,t^{\text{open}}}}{p_{j,t^{\text{open}}}} \right) - \mathcal{C} \times (|v_{i,t}| + |v_{j,t}|), \qquad (22)$$

where t^{open} is the timestamp of the latest open position to t. It is worth noting that the difference of stock prices (e.g., $p_{i,t} - p_{i,t^{\text{open}}}$) is normalized to the corresponding stock price at the open position (e.g., $p_{i,t^{\text{open}}}$). The reason is twofold: (1) If it is not normalized, those pairs with greater differences would be more preferred after learning. However, they could also bring greater risks. (2) It is more fair to put equal attention on those pairs with smaller differences but great normalized differences since they are profitable as well. It is possible that SADQN can earn great profit by elaborating the trading volume (i.e., $v_{i,t}$ and $v_{j,t}$). Overall, the net return is the normalized profit generated by a transition from the current state s^t to the next state s^{t+1} via action a and the transaction cost has been already deducted. However, the original definition of net profit [50] in Eq. (22) are not aware of risks. In this regard, we categorize three different close conditions: normal close, stop-loss close, and exit. We detail them and define the risk-aware reward $R(s^t, a^t, s^{t+1})$ for each condition as follows:

- **Normal close**: Normally, a positive profit is returned when the spread is closed at its historical mean in trading periods. Besides, to avoid risky situations, it is more

encouraged to close positions at where the risks of structural breaks and market close are low. Accordingly, we formulate the reward of *normal close* as below:

$$R(s^t, a^t, s^{t+1}) = 1000 \times NR_{i,j}^t \times r_t \times (1 - Pr(i,j,t)), \tag{23}$$

where the complement of occurrence probability of structural break $1 - Pr(i,j,t)$ and market-closing risk r_t discount the net return $NR_{i,j}^t$. As a result, the reward of *normal close* is great if the net profit is great and both the risks are low. 1000 is a constant, following [50], denoting the profitability of normal close. Generally, this type of reward is positive.

- **Stop-loss close**: When the spread diverges beyond expectation, it is necessary to close positions immediately to avoid further loss. In this case, the reward is set to negative since it usually causes great loss. The reward *stop-loss close* is defined as follows:

$$R(s^t, a^t, s^{t+1}) = -1000 \times \left| NR_{i,j}^t \right| \times (1 - r_t) \times Pr(i,j,t), \tag{24}$$

where the absolute net return measures the amplitude of loss. 1000 is a constant similar to normal close. Under this formulation, when the estimated risks increase, the negative reward is amplified to alert the pairs trading strategy to stay away from such dangerous situations.

- **Exit**: The trades are forced to be closed (namely exit) when the trading period ends every day. However, it is hard to guarantee the profit is positive when exit happens. Due to the dynamic trends, the return could be either positive or negative. Therefore, the reward of *exit* is defined as below:

$$R(s^t, a^t, s^{t+1}) = \begin{cases} 500 \times NR_{i,j}^t \times r_t \times (1 - Pr(i,j,t)), & \text{if } NR_t > 0 \\ -500 \times \left| NR_{i,j}^t \right| \times (1 - r_t) \times Pr(i,j,t), & \text{otherwise}. \end{cases} \tag{25}$$

By following [50], the constant term (i.e., ± 500) is set to half of that in the other close conditions because exit usually causes less profit, no matter it is positive or negative.

Although structural breaks are risky and are likely to cause tight boundaries, the position of spread could be still far away from the stop-loss boundaries and hence terminate with normal close or exit later. Besides, SAPT comprehensively considers multiple risk features, including the structural break probabilities, market-closing risk, etc. Therefore, there could be a chance that SAPT still considers the trading environment is safe with structural breaks when other indicators report safe, and SAPT decides to fix the boundaries accordingly.

Equipped with the above definitions of MDP, given the current state s^t at time t, SADQN aims at selecting the best action a^t of maximum Q-value $Q^*(s^t, a^t)$, which is defined as the sum of expected reward as follows:

$$Q^*(s^t, a^t) = E_{s^{t+1}}[R(s^t, a^t, s^{t+1}) + \gamma \max_{a^{t+1}} Q(s^{t+1}, a^{t+1})|s^t, a^t], \tag{26}$$

where $\gamma \in [0, 1]$ is a factor discounting the maximum possible Q-values in the future. To approximate the Q-value, SADQN stacks several fully connected layers to learn and extract sophisticated information, as illustrated in the yellow rectangle in Fig. 4. For simplicity, the neurons in the fully connected layers are gathered to be the learnable parameter set θ. Accordingly, the loss function to approximate the Q-value is defined as follows:

$$\mathcal{L}(\theta) = (R(s^t, a^t, s^{t+1}) + \gamma \max_{a^{t+1}} Q(s^{t+1}, a^{t+1}) - Q(s^t, a^t; \theta))^2. \tag{27}$$

That is, we aim to approximate the original Q-value $Q(s^t, a^t; \theta)$ to the target Q-value $Q^*(s^t, a^t)$ by learning the parameters θ. In each iteration of the training phase, the original Q-value $Q(s^t, a^t)$ of each corresponding action a^t and s^t is first derived from the fully connected layers and then is used to, respectively, update θ and Q-value $Q^*(s^t, a^t)$ as follows:

$$\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta) \tag{28}$$

$$Q(s^t, a^t) \leftarrow Q(s^t, a^t) + \eta[R(s^t, a^t, s^{t+1}) + \gamma \max_{a^{t+1}} Q(s^{t+1}, a^{t+1}) - Q(s^t, a^t)], \tag{29}$$

where η is the learning rate. The learning parameters in θ are learned by the gradient descent. The time complexity of Eq. 28 is $O(|A| + |s^t|)$, where $|A|$ and $|s^t|$ denote the number of actions and the dimension of a state. The new Q-value is updated by the sum of the current Q-value $Q(s^t, a^t)$ and the difference between the current Q-value $Q(s^t, a^t)$ and the expected maximum Q-value $R(s^t, a^t, s^{t+1}) + \gamma \max_{a^{t+1}} Q(s^{t+1}, a^{t+1})$. The time complexity of updating Q-value is $O(|A|)$, which is much smaller than conventional Q-learning ($O(|S| \times |A|)$) asymptotically. The training process stops until the loss is converged or a predefined maximum iteration is met. Finally, in the inference stage, given a state s^t, the action $a^t \in A$ with the maximum derived Q-value is the output of SADQN.

6 Experiments

In this section, we describe the large-scale dataset used in Sect. 6.1. The experimental setup and results of structural break detection are, respectively, presented in Sects. 6.2 and 6.3; the experimental setup and results of learning pairs trading strategy are detailed in Sects. 6.4 and 6.5. Finally, real case studies in special environment of stock markets, including the impact of coronavirus (COVID-19) and the volatile market, are provided as well. For further experiments, we conduct the experiments on a workstation equipped with two Intel E5-2683 V3 CPUs, a Titan X Pascal graphics card and 189 GB main memory, and the entire proposed framework is implemented by Python with Keras and Keras-RL.[3]

---

[3] Keras-RL, https://github.com/keras-rl/keras-rl.
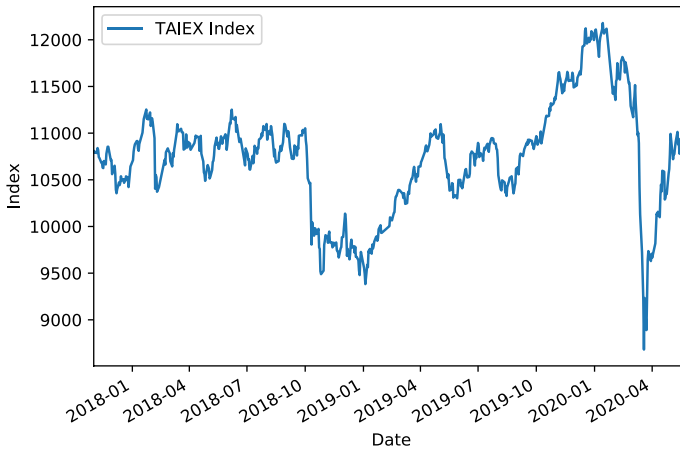
Fig. 7 Index of TAIEX from 2018/1 to 2020/5
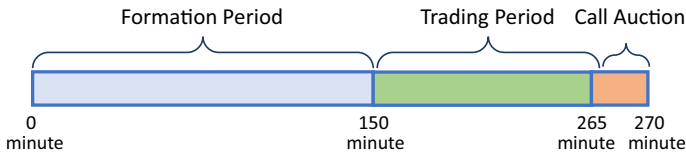
Fig. 8 Illustration of formation and trading period in the Taiwan stock market

6.1 Data preparation

We collect the stock tick data from the top 150 companies in the Taiwan Stock Exchange Capitalization Weighted Stock Index (TAIEX)[4] to ensure the liquidity. The data are collected from November 1st, 2017 to May 31st, 2020. Following [61], to approximate the real market environment, stock prices are aggregated to volume-weighted average prices in minute scale. Figure 7 illustrates the index of the TAIEX dataset. It is worth noting that TAIEX includes important trends, such as bear, bull, and oscillating markets, during this period.

Taiwan stock market opens from 09:00 to 13:30, where the last 5 min before close will not have any trade due to the call auction mechanism. As illustrated in Fig. 8, in the first 150 min, we apply the VECM [28] to extract the cointegrated pairs every day. There are 19,872 cointegrated pairs extracted in total. The following 115 min are the interval for trading.

---

[4] Taiwan Stock Exchange Corporation (TWSE), https://www.twse.com.tw/en/.

6.2 Experiment setup for structural break detection

Since labeling ground truth of breakpoints requires experienced experts and is time-consuming, we follow [41] to obtain breakpoints with [54] developed by Yahoo. 75% and 25% of the data are, respectively, used for training and testing, and the last 20% of the training data are for validation. All the time series, including stock prices and spreads, are normalized to eliminate offsets by following [11].

6.2.1 Evaluation metrics

To evaluate the performance, we define that a true breakpoint at time bp is detected whether the breakpoint alerted by an algorithm is in the time interval $[bp - \tau_1, bp + \tau_2]$, where τ_1 and τ_2 are given tolerances ranging from 0 to 70 min. In general, setting a great τ_1 is more conservative (sensing structural breaks early) and setting a great τ_2 is more aggressive (allowing late responses to structural breaks). Therefore, the setting of τ_1 and τ_2 can depend on the personality or the strategy of the agents. In our case, since SAPT has already considered the risk factors in its state function (potentially making SAPT conservative), we set τ_1 to 0 in order to maximize the profitability. The evaluation metrics are listed as follow:

- **True Detection Rate**: the ratio of the number of detected breakpoints to the number of all breakpoints.
- **Partial Detection Rate** [51]: the ratio of the number of breakpoints that are alerted earlier than bp to the number of all breakpoints.
- **Missed Rate**: the ratio of the number of non-detected breakpoints to the number of all breakpoints.
- **False Detection Rate** [51]: the ratio of the number of spreads without breakpoints but alerted by an algorithm to the number of spreads without breakpoints.
- **Average Delay**: the average time difference between the timestamp of the detected breakpoint and the timestamp of the actual breakpoint (ground truth).

6.2.2 Comparison methods

We compare SWANet with statistical methods, change-point detection methods, and a variant of SWANet:

- **3-std** [17, 32, 43]: In pairs trading, finance experts often alert structural breaks if the spread exceeds a predefined threshold. Following [32, 43], we derive the standard deviation of each spread in the formation period every day and the threshold is set to three standard deviation of each spread.
- **ADF** [25]: Augmented Dickey–Fuller test is a widely used statistical method to test whether a time series is stationary or not. Following the original litera-
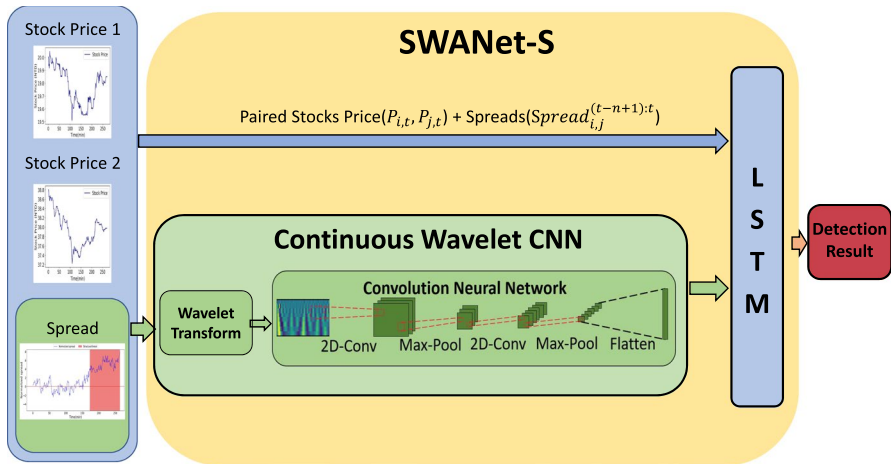
Fig. 9 SWANet-S architecture

ture, ADF obtains its stationary parameters in formation period every day and returns a breakpoint whenever the spread becomes non-stationary in trading period.

- **BCD** [1]: Bayesian online change-point detection is a popular method to detect abrupt changes in time series data online. A breakpoint is reported by BCD if it is identified as a change point.
- **LSTM** [40]: Long short-term memory is the state-of-the-art recurrent neural network (RNN) method for time series analysis. The inputs of LSTM are the stock prices and the spreads of cointegrated pairs, which are identical to the descriptions in Sect. refsec:hybrid.
- **SWANet**: The method proposed in this paper. The continuous wavelet CNN and LSTM are combined in parallel, and a fully connected layer is responsible for integrating their information to derive the occurrence probability of structural breaks. The architecture is shown in Fig. 5. If the derived occurrence probability is greater than 90%, than a breakpoint is returned (detailed later). Comparing with LSTM, we show the power of integrating continuous wavelet CNN and LSTM.
- **SWANet-S**: SWANet-S is a variant of SWANet. The main differences are that SWANet-S connects the continuous wavelet CNN and the LSTM in series and the final result is directly determined by the LSTM. The architecture is shown in Fig. 9. Similarly, if the occurrence probability is greater than 90%, a breakpoint is returned. Comparing with SWANet-S allows us to study which way to combine is better.

Note that the statistical methods, 3-std and ADF, detect structural breaks offline, which is not ideal for real cases. The other methods are tested online.

Table 2 Performance of each method (τ min)

	Miss rate (%)	True detection rate (%)	Partial detection rate (%)	False detection rate (%)	Average delay (min)
3-std	30.5	24.4	19.6	44.9	22.310
ADF	27.4	15.6	39.3	54.8	24.729
BCD	52.8	11.4	24.5	34.6	22.342
LSTM	22.4	44.7	3.6	5.7	18.284
SWANet-S	**18.4**	46.0	8.4	13.2	17.319
SWANet	21.0	**47.3**	**3.2**	**5.1**	**17.315**

The best method in each case/experiment is marked as bold. It is a convention of the machine learning field

6.3 Experimental results of structural break detection

Table 2 compares multiple evaluation metrics of each method. SWANet and SWANet-S outperform the other baselines in all the metrics, manifesting that learning from both frequency- and time-domain features are important for structural break detection. The improvement in partial detection rate and false detection rate, which are 61.3% and 61.9%, respectively, are especially significant compared to traditional methods. The results explain that SWANet seldom makes false detection in cointegrated pairs, thereby keeping more arbitrage opportunities for pairs trading strategies.

SWANet improves LSTM by at least 5.3% in all metrics because SWANet extracts sophisticated frequency-domain features from scalograms with two-dimensional convolution layers additionally. To have a closer look, Fig. 10 shows the distribution of delay of all methods. The timestamps of detected breakpoints by SWANet and SWANet-S are closer to the real breakpoints compared to LSTM. Figure 11 depicts the distribution of output occurrence probability of structural breaks of SWANet. The output is clearly separated into two clusters, where the one with higher probability ($\geq$ 90%) represents those are suspect breakpoints returned by SWANet. As a result, we recommend to set the threshold to be 90% in TAIEX dataset.

For sensitivity tests, Fig. 12 shows the true detection rates over different delay tolerance τ_2 in testing data. Neural network-based methods are significantly better than the others because they are capable of extracting effective features from time series data. When the delay tolerance is small (i.e., 10 min), SWANet significantly outperforms those statistical methods by at least 182%, showing stronger potential in use of real stock markets. Moreover, SWANet improves SWANet-S by at least 2.6%, which manifests that the fully connected layers integrates the parallel combination of continuous wavelet CNN and LSTM successfully, rather than concatenating them in series. As mentioned in Sect. 2, ADF and BCD perform
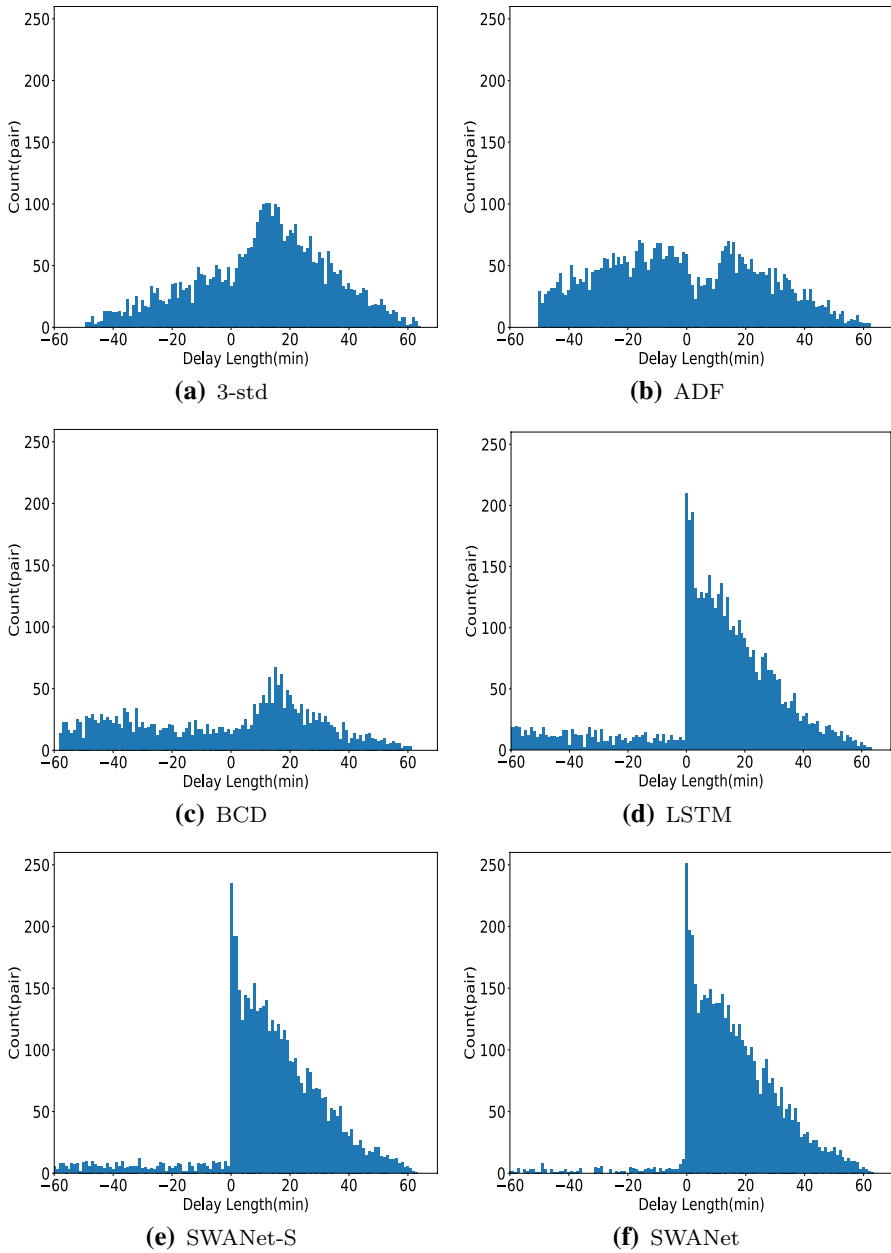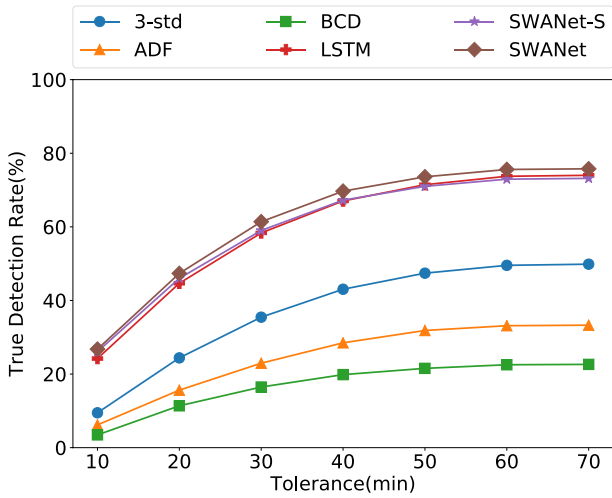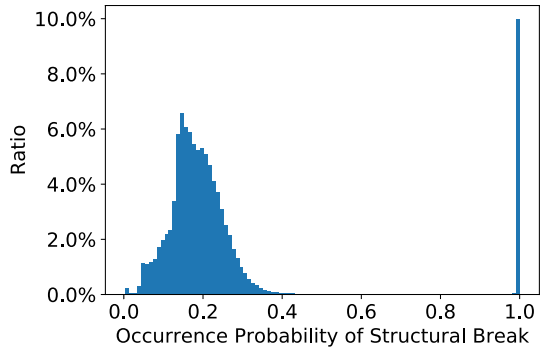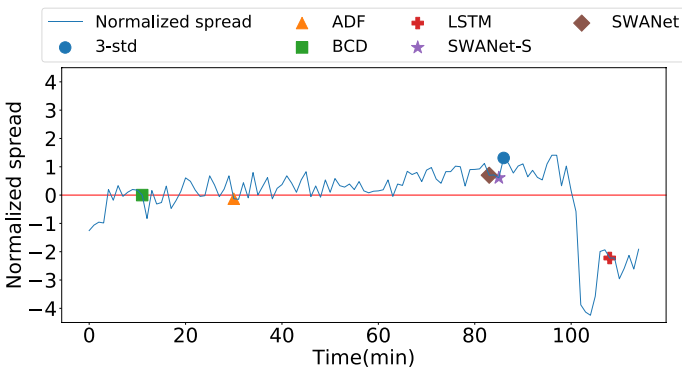
(a) 3-std

(b) ADF

(c) BCD

(d) LSTM

(e) SWANet-S

(f) SWANet

Fig. 10 Delay distribution

Fig. 11 Occurrence probability of structural break

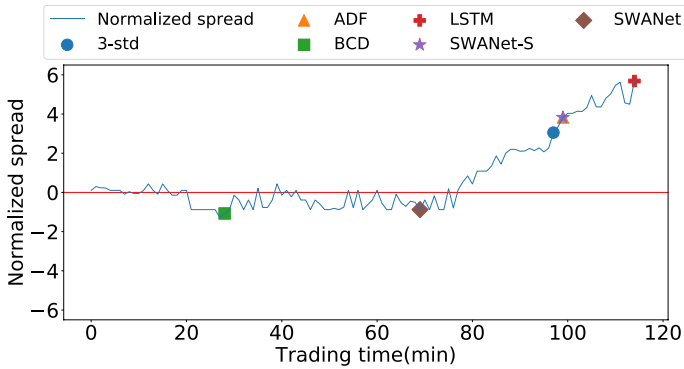**Fig. 12** True detection rate of each method

Fig. 13 Case 1
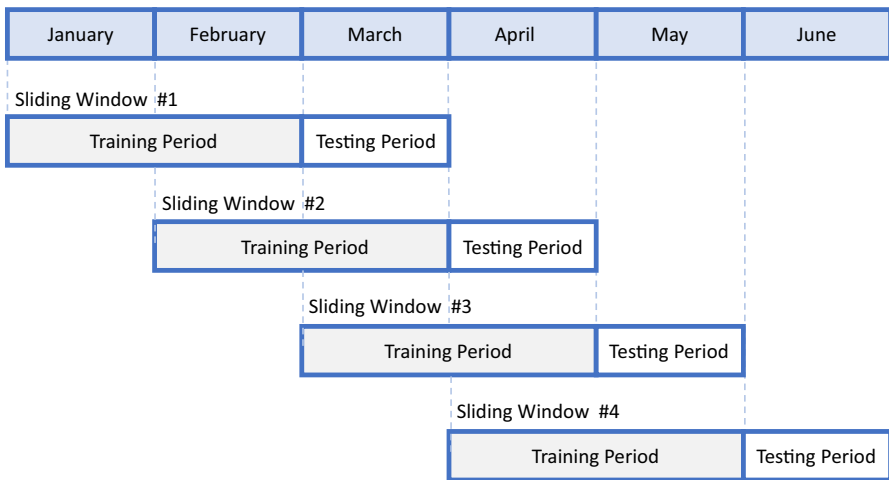
Fig. 14 Case 2

Fig. 15 An example of training and testing with sliding window

worse than the 3-std because of the high sensitivity of detection in high fluctuation and variety in the financial market.

6.3.1 Case study

We pick two cases which have obvious structural breaks to explain the performance of each method. As shown in Fig. 13, the first case is the pair of stock id 2308 and 2439 on May 22nd, 2019, in which the breakpoint roughly locates at the time interval from the 80th minute to the 90th minute. BCD and ADF are too sensitive such that they detect the breakpoint early. In contrast, LSTM is too late (roughly 20 min) to detect the structural break. SWANet, SWANet-S, and 3-std perform well in this case. Figure 14 shows the second case, which is the pair of

stock id 2492 and 2912 on May 24th, 2019. The breakpoint also roughly locates at the time interval from the 70th minute to the 90th minute. SWANet is the only method to find the right position of the breakpoint in this case.

6.4 Experiment setup for pairs trading strategy optimization

6.4.1 Simulation

To mitigate the influence of slippage in TAIEX, we set the maximum trading shares of each stock to 5000 and initial capital to 10 million New Taiwan dollars (TWD) [53]. Following [22], we construct the training and testing data with sliding windows as shown in Fig. 15. To be more specific, the transactions in two months are used for training and those in the following one month are used for testing. The last two weeks in training data are used for validation. Consequently, all the models learn from recent market trends. The default transaction cost, following Taiwan's law, is the transaction tax, which is 0.15% of the stock price. Among most of the major stock markets in the world, the transaction cost in Taiwan stock market is the highest. All the models play the roles of dealers and follow the day trading strategy in the simulation.

6.4.2 Evaluation metrics

We evaluate the performance of each pairs trading strategy from two crucial and widely used aspects: profit and risk. The details are listed as follows:
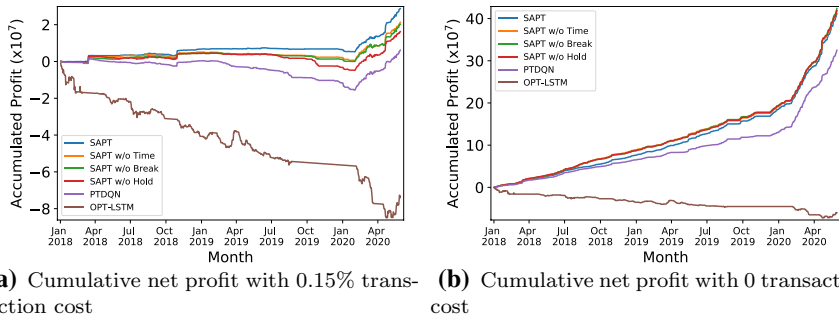
- **Indicators of Profit**:
- **Cumulative net profit**: The cumulative profit of the investment *after* subtracting transaction cost.
- **Indicators of Risk** [70]:
- **Maximum drawdown (MDD)**: The drawdown is the measurement of downside risk from a peak in the account balance *AccBalance*, where *AccBalance* is the sum of cumulative net profit and initial capital. Maximum drawdown is defined as:

$$\text{MDD(T)} = \max_{\tau \epsilon (0,T)} \left[\max_{t \epsilon (0,\tau)} \frac{AccBalance(t) - AccBalance(\tau)}{AccBalance(t)} \right] \quad (30)$$

  Note that greater MDD represents higher risks.

- **Sharpe ratio**: With risk consideration, Sharpe ratio is evaluated by average return in excess of risk-free rate under volatility of risk. Sharpe ratio is formulated as:

$$\text{Sharpe ratio} = \frac{R_p - R_f}{\sigma_p}, \quad (31)$$

(a) Cumulative net profit with 0.15% transaction cost

(b) Cumulative net profit with 0 transaction cost

Fig. 16 Feature influence on cumulative net profit from 2018/1 to 2020/5

where R_p is the average of daily returns. R_f is the risk-free rate, which is set to 0 as [50]. σ_p is the standard deviation of the daily returns which measures the volatility.

- **Sortino ratio**: By using the standard deviation of negative profit instead of total profit, Sortino ratio evaluates the excess return under downside deviation. Sortino ratio is formulated as:

$$\text{Sortino Ratio} = \frac{R_p - R_f}{\sigma_{dp}}, \tag{32}$$

where R_p and R_f are the same as Sharpe ratio, and σ_{dp} is the downside deviation of the daily returns which measures the downside volatility.

6.4.3 Comparison methods

- **PTDQN**: PTDQN [50] is a deep Q-network which learns pairs trading strategies by optimizing trading and stop-loss boundaries, but it only considers the spread as the state and adopts the return as the reward in deep Q-network.
- **OPT-LSTM** [71]: OPT-LSTM applies an unsupervised OPTICS clustering algorithm to select pairs and conducts LSTM to predict spread based on its trend. Following the original literature, the trading boundaries are fixed to one standard deviation of each spread and no stop-loss boundary is included.
- **SAPT**: The proposed method which factors in the occurrence probability of structure break, the market-closing risk, the transaction cost with a *hold* action.
- **SAPT w/o Break**: SAPT without considering the structure break probability.
- **SAPT w/o Time**: SAPT without considering the market-closing risks.
- **SAPT w/o Hold**: SAPT without the *hold* action. In other words, unlike SAPT, this method does not trade whenever it meets a trading or stop-loss boundary.
- **SAPT-3-std**: SAPT using the structure break probability predicted by 3-std.
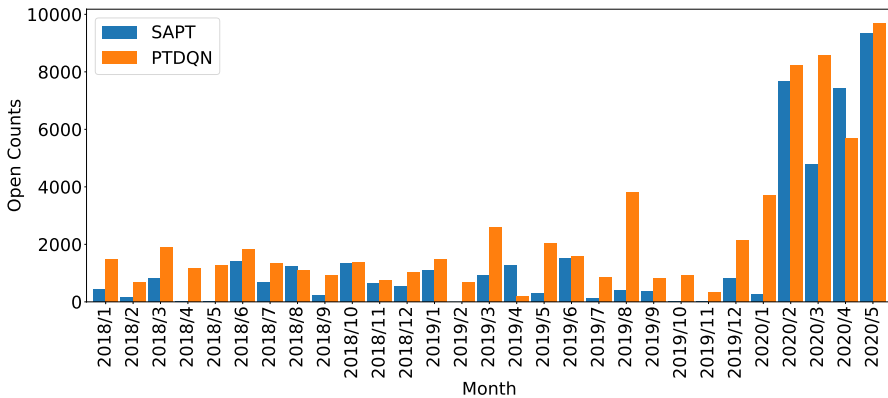
Fig. 17 Trade volume of SAPT and PTDQN in each month from 2018/1 to 2020/5

Table 3 Trade counts and net profit of SAPT and PTDQN in 2020/5		Normal close count	Stop-loss close count	Exit close count	Profit
	SAPT	7656	461	267	743,485
	PTDQN	8342	342	358	647,546

- **SAPT-ADF**: SAPT using the structure break probability predicted by ADF.
- **SAPT-BCD**: SAPT using the structure break probability predicted by BCD.
- **SAPT-LSTM**: SAPT using the structure break probability predicted by LSTM.

6.5 Experimental results of pairs trading strategy optimization

6.5.1 Performance analysis

Figure 16 presents the cumulative net profit of each method with different transaction cost in the testing dataset. PTDQN and OPTICS-LSTM perform worse because they do not optimize a transaction cost-aware objective nor the cost-aware reward function, and hence, the profit is consumed by the transaction cost at 0.15%. Moreover, OPTICS-LSTM fixes its cointegrated pairs after training, which does not reflect the bumpy intraday trading market. SAPT outperforms the others by at least 77%, manifesting that all of the three modules are crucial in optimizing pairs trading strategies. To be more specific, the influence of the three modules is stronger when the transaction cost is higher, evident by that SAPT improves the cumulative net profit by 77% with 0.15% transaction cost. On the other hand, compared to the other variants, SAPT w/o Hold, respectively, increases at least 15% and 25% of the number of trades and unprofitable trades, leading to the least profit in high-transaction-cost environment (0.15%). The result shows the effectiveness of the hold action to filter

Table 4 Risk indicators of each method

	Sharpe ratio	Sortino ratio	MDD
SAPT	**4.30**	**13.18**	**0.020**
SAPT w/o Break	3.45	9.53	0.044
SAPT w/o Time	3.42	9.78	0.043
SAPT w/o Hold	3.07	6.71	0.090
PTDQN	1.01	1.41	0.169
SAPT-3-std	1.07	1.77	0.143
SAPT-ADF	− 0.23	− 0.32	0.127
SAPT-BCD	− 3.15	− 2.95	0.250
SAPT-LSTM	− 1.32	− 1.49	0.297

The best method in each case/experiment is marked as bold. It is a convention of the machine learning field

out unprofitable open positions as well as to reduce the amount of trades. However, SAPT w/o Hold outperforms SAPT in low-transaction-cost environments (0% transaction cost) for the cumulative net profit by 1.7%, as shown in Fig. 16b. This is because SAPT w/o Hold improves the number of profitable trades by 39.3% from the environment of 0.15% to 0% transaction cost, whereas the number of unprofitable trades are increased only 2.9%. Compared to SAPT w/o Hold, SAPT looks for steady profit even in the low-transaction-cost environment by incorporating the hold action to alleviate risks and hence losing those very profitable but risky arbitrage opportunities.

Figure 17 illustrates the total number of transactions for the positions opened every month, where the blue and the orange bars are the trading volume of SAPT and PTDQN, respectively. Overall, SAPT is more conservative than PTDQN since the total amount of open positions of SAPT is fewer (43,772 vs. 68,161). However, combined with the results in Fig. 16a that SAPT actually earns more profit, SAPT strikes a better balance between risks and profit than PTDQN. On the other hand, the difference of total open counts between SAPT and PTDQN during the pandemic period of COVID-19 (2020/2 to 2020/5) is far less than that not in the pandemic period (10.1% vs. 147%). It shows that SAPT becomes aggressive during the pandemic by adjusting the trading boundaries and invoking less hold action such that opening profitable trades is easier. The average monthly profit of SAPT during the pandemic thus increases 143%. That is, SAPT is adaptive to market environment.

To examine the reward function designed for the normal close, stop-loss close and exit, we compare the behavior of SAPT and PTDQN in May 2020, as shown in Table 3. Surprisingly, the amount of normal close of SAPT is 8.2% less than PTDQN, which is because PTDQN does not take the transaction cost into consideration, and hence, several unprofitable positions are triggered. In contrast, SAPT invokes more profitable normal close than PTDQN by 2.3%, which further increases 8.2% profit of each trade with normal close in average. On the other hand, SAPT has 34.8% more stop-loss close than PTDQN in terms of amount since SAPT includes the market-closing risk and the occurrence of structural break in the reward function
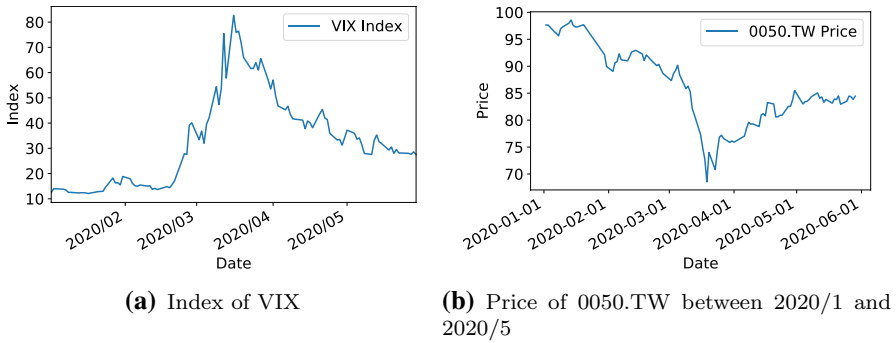
(a) Index of VIX

(b) Price of 0050.TW between 2020/1 and 2020/5

Fig. 18 VIX index and 0050.TW price during COVID-19

(a) TAIEX index between 2020/1 and 2020/5
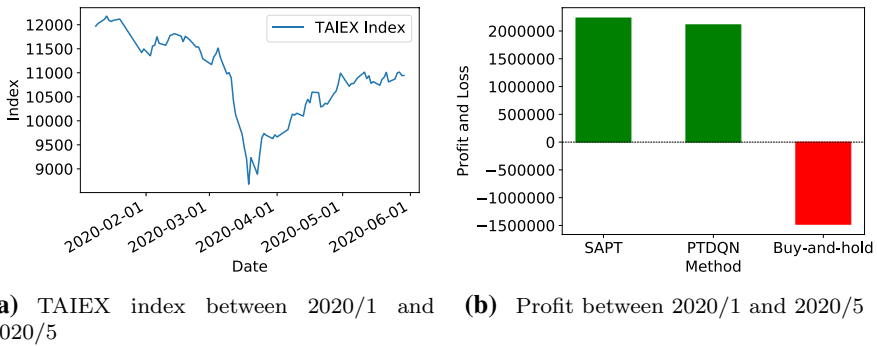
(b) Profit between 2020/1 and 2020/5

Fig. 19 Pairs trading during COVID-19

to avoid trading at high-risk positions. In addition, by considering the market-closing risk, SAPT decreases 25.4% of the amount of exit, leading to 38.7% reduction of the total loss caused by exit. Overall, the cost- and risk-aware reward function significantly changes SAPT 's behavior and effectively improves the total profit by 14.8%.

From the perspective of risk control, the upper part of Table 4 shows the Sharpe ratio, Sortino ratio, and MDD of all the pairs trading methods. SAPT outperforms the others by at least 24.7% among all the three indicators, manifesting that SAPT controls the risks effectively. It is worth noting that SAPT has the greatest improvement on Sortino ratio comparing with others (34.7%), showing significant profitability while the market is generally suffering from a decline (i.e., downside risks). SAPT w/o Hold has the worst MDD, Sharpe ratio, and Sortino ratio among its variants, which means removing the hold action loses the mechanism to filter out unprofitable trades. However, all of the variants of SAPT outperform PTDQN by at least 46% in all risk indicators, manifesting that the proposed risk control mechanisms are effective.

In the lower part of Table 4, we study the performance of SAPT by adopting different structural break prediction methods. The proposed SAPT using

(a) TAIEX index on March 17, 2020

(b) Profit on March 17, 2020
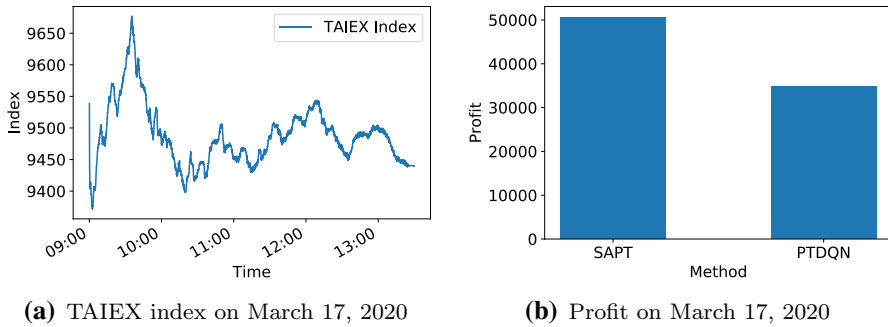
Fig. 20 Pairs trading in high-volatility market on March 17, 2020

SWANet significantly outperforms the other baselines among all risk indicators. While 3-std and ADF are statistical methods that do not require labeled structural breaks, they have shown significant declines of all risk indicators, compared to SAPT. Moreover, SAPT-ADF even has negative Sharpe ratio and Sortino ratio, since its revenue per trade is negative (-262.04 TWD per trade). This result also qualifies our simulated labels and the proposed structural break detection method SWANet.

6.5.2 Case study: the market panic caused by coronavirus disease 2019

Due to the outbreak of the coronavirus disease 2019 (COVID-19) which had brought up uncertainty and panic to the market and even triggered circuit breaker in the stock market all over the world. Chicago Board Options Exchange Volatility Index (VIX Index) had risen drastically in March 2020 as shown in Fig. 18a, manifesting that the market is full of panic and is expecting high volatility [72]. In our dataset, TAIEX had fallen over 3500 points (28.7%) from January to March, and had risen 2500 points (26.8%) back from late March to May, as shown in Fig. 19a. To justify the pairs trading strategy, the commonly used buy-and-hold strategy is adopted for comparison. Since TAIEX is not an instrument for trading, the Taiwan 50 ETF (0050. TW)[5] had a similar trend as TAIEX as shown in Fig. 18b, which had been used to serve as the buy-and-hold trading target. Figure 19b shows the profit of different methods in test data. SAPT has higher profit comparing with PTDQN, while buy-and-hold method suffers from a great loss. It is because pairs trading has the characteristic to offset the system risks in the market, and hence, it is less likely to suffer from a great loss compared to the buy-and-hold method. Moreover, there exist more divergences of spread in volatile markets, resulting in more arbitrage opportunities.

---

[5] TAIEX includes the stocks with top 50 market value and it is accounted more than 70% of the total market value in Taiwan stocks [45].
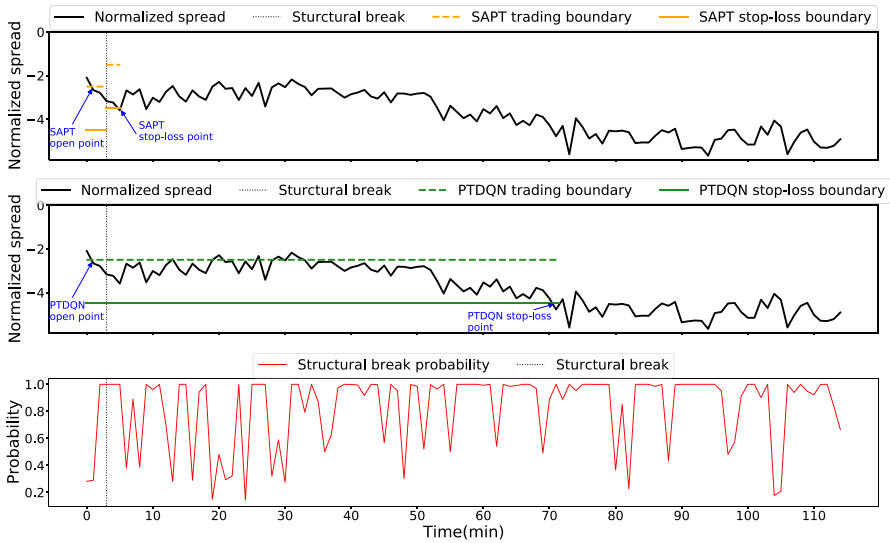
Fig. 21 A case of SAPT (top) and PTDQN (middle) undergoing structural breaks (bottom). Only the trading period is displayed

6.5.3 Cases study: volatile market

The economic growth slowed down due to the COVID-19 pandemic. The Federal Reserve System (FED) announced to reduce interest rate and the pessimism in the future market[6] even triggered the circuit breaker in US stock market in the next trading day. As the Taiwan stock market is highly related to the international financial environment [45], there is an intensive fluctuation in the Taiwan stock market on March 17, 2020. TAIEX dropped 279 points (2.9%) during the trading period, as shown in Fig. 20a, which is the trading day with the highest volatility in our dataset. Moreover, the VIX index, which indicates how panic the market is, even came to a peak as shown in Fig. 18a. Figure 20b compares the daily net profit of SAPT and PTDQN on March 17. SAPT outperforms PTDQN by 45%, manifesting that SAPT has greater hedging ability in highly volatile market. This is because SAPT takes the market factors into consideration including structural break probability, transaction cost and time remain ratio. The trade count of SAPT has dropped 45% comparing with PTDQN, which shows the effectiveness of selecting better decision and timing to trigger trades with better return.

---

[6] Federal Reserve slashes interest rates to zero as part of wide-ranging emergency intervention,https://www.washingtonpost.com/business/2020/03/15/federal-reserve-slashes-interest-rates-zero-part-wide-ranging-emergency-intervention/.
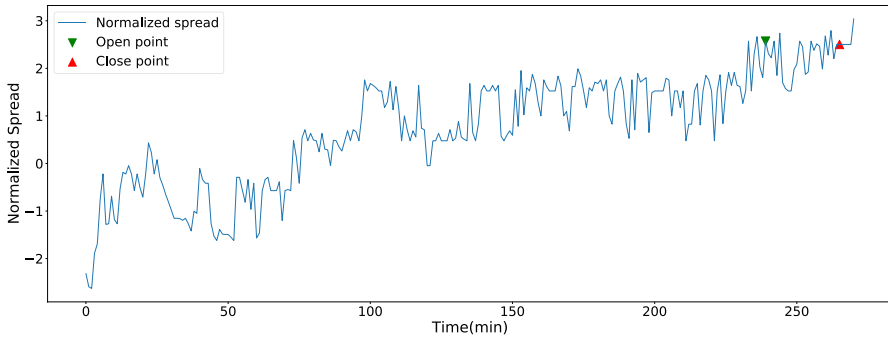
Fig. 22 A case of forced close by exit

6.5.4 Case study: how SWANet affects SADQN

To investigate the effectiveness of combining SWANet and SADQN, Fig. 21 shows the trading period of a cointegrated pair of stocks Delta Electronics, Inc. (2308.TW) and Quanta Computer Inc. (2382.TW) on July 27, 2018. The top and the middle plots show the policies of SAPT and PTDQN, respectively. The orange and the yellow lines indicate their strategies, where the solid lines and the dotted lines represent the stop-loss and the trading boundaries, respectively. The bottom plot presents the occurrence probability of structural break predicted by SWANet. In the beginning of the trading period (time 0), SAPT and PTDQN have identical boundaries. They both open at time 2 while the spread (black solid lines) meets the trading boundaries. Based on the principle of pairs trading strategy, both of them expect the spread will converge to mean (i.e., zero in a normalized spread) so they can make profit between the gap of the open positions and the mean.

However, the policy of SAPT suddenly changes (i.e., the trading and the stop-loss boundaries shift) at time 5 (gray vertical dotted line) and the trade is closed at time 6 since the spread meets the new stop-loss boundary. This sudden change credits to the high occurrence probability of structural break starting at time 4, as shown in the bottom plot. SAPT senses the structural break (i.e., fading away from the mean) so it tightens its boundaries immediately to avoid unexpected loss. On the other hand, PTDQN does not aware of the risk of structural breaks so it does not change its boundaries accordingly. In this case, the trade is closed until time 72, when the spread meets a wider stop-loss boundary, compared to SAPT. Therefore, the loss of PTDQN is roughly 150% of the loss of SAPT.

6.5.5 Case study: market-closing risk

Figure 22 illustrates the trading actions of PTDQN on July 23, 2019 with the cointegrated pair of stocks Delta Electronics, Inc. (2308.TW) and Formosa Petrochemical Corp. (6505.TW). PTDQN opens at time 240 (green triangle). However, it does not aware that the market is closing at time 265, and hence is forced to exit (red triangle)

according to Taiwan's law[7]. This forced close results in -6.8% return of invest- ment. On the other hand, SAPT includes the *market-closing risk* r_t in the state set of SADQN, making SAPT aware the risk of exit and trigger the *hold* action. Unlike PTDQN, SAPT avoids the loss.

7 Conclusion and future work

To the best of our knowledge, there is no prior research considering risk controls, including structural breaks, and transaction costs in optimizing pairs trading strat- egy. In this paper, to tackle this urgent need, we propose a two-phase framework SAPT. The first-phase SWANet detects structural breaks by extracting not only time-domain features from stock price and spread with an LSTM but also fre- quency-domain features from scalograms by using a continuous wavelet CNN. The second-phase SADQN optimizes the structural break-aware pairs trading strategy with a deep Q-network. Via experimental results and real case studies in the large- scale dataset TAIEX in Taiwan stock market, we show that: (1) SWANet outper- forms conventional statistical methods by at least 93.9% in terms of true detection rate, (2) SADQN, respectively, outperforms the other state-of-the-art methods by at least 456% and 934% in terms of profit and Sortino ratio, and (3) SAPT is robust in volatile market environment caused by COVID-19 pandemic. For the future work, while there are 1730 stocks potentially generating about 1.5 million pairs in Taiwan stock market in 2020, monitoring all the pairs in real time requires huge compu- tation resources. Therefore, federated learning and distributed learning frameworks that distribute partial data to multiple resources are potentially promising to address the needs. Nevertheless, how to strike a good balance between the scalability and the performance (e.g., cumulative profit) learned from partial data should be care- fully examined.

References

1. Adams RP, MacKay DJ (2007) Bayesian online changepoint detection. arXiv preprint arXiv:0710. 3742
2. Alvey B, Ho DK, Zare A (2017) Fourier features for explosive hazard detection using a wide- band electromagnetic induction sensor. In: Detection and sensing of mines, explosive objects, and obscured targets XXII, vol. 10182. International Society for Optics and Photonics, p 101820E
3. Aminikhanghahi S, Cook DJ (2017) A survey of methods for time series change point detection. Knowl Inf Syst 51(2):339–367

---

[7] In Taiwan's stock market, time 265–270 is for calling auctions, which do not support pairs trading.

4. Aminikhanghahi S, Wang T, Cook DJ (2018) Real-time change point detection with application to smart home time series data. IEEE Trans Knowl Data Eng 31(5):1010–1023

5. Andrade S, Di Pietro V, Seasholes M (2005) Understanding the profitability of pairs trading. Unpublished working paper, UC Berkeley, Northwestern University

6. Ankirchner S, Dimitroff G, Heyne G, Pigorsch C (2012) Futures cross-hedging with a stationary basis. J Financ Quant Anal 66:1361–1395

7. Bader DA, Agarwal V (2007) Fftc: fastest Fourier transform for the IBM cell broadband engine. In: International Conference on High-Performance Vomputing. Springer, pp 172–184

8. Bao W, Yue J, Rao Y (2017) A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLoS ONE 12(7):e10180944

9. Bellman R (1957) A Markovian decision process. J Math Mech 66:679–684

10. Bento P, Pombo J, Calado M, Mariano S (2019) Optimization of neural network with wavelet transform and improved data selection using bat algorithm for short-term load forecasting. Neurocomputing 358:53–71

11. Bowen D, Hutchinson MC, O'Sullivan N (2010) High-frequency equity pairs trading: transaction costs, speed of execution, and patterns in returns. J Trad 5(3):31–38

12. Box GE, Jenkins GM, Reinsel GC (2011) Time series analysis: forecasting and control, vol 734. Wiley

13. Bu L, Babu R, De Schutter B et al (2008) A comprehensive survey of multiagent reinforcement learning. IEEE Trans Syst Man Cybernet C Appl Rev 38(2):156–172

14. Caldeira J, Moura GV (2013) Selection of a portfolio of pairs based on cointegration: a statistical arbitrage strategy. Available at SSRN 2196391

15. Cao J, Li Z, Li J (2019) Financial time series forecasting model based on ceemdan and LSTM. Phys A Stat Mech Its Appl 519:127–139

16. Chen CW, Chen M, Chen SY (2014) Pairs trading via three-regime threshold autoregressive Garch models. In: Modeling dependence in econometrics, pp 127–140

17. Chen D, Cui J, Gao Y, Wu L (2017) Pairs trading in Chinese commodity futures markets: an adaptive cointegration approach. Acc Finance 57(5):1237–1264

18. Chen G, Lu G, Shang W, Xie Z (2019) Automated change-point detection of EEG signals based on structural time-series analysis. IEEE Access 7:180168–180180

19. Chen YY, Chen WL, Huang SH (2018) Developing arbitrage strategy in high-frequency pairs trading with filterbank CNN algorithm. In: IEEE International Conference on Agents (ICA). IEEE, pp 113–116

20. Chow GC (1960) Tests of equality between sets of coefficients in two linear regressions. Econom J Econom Soc 66:591–605

21. Cummins M, Bucca A (2012) Quantitative spread trading on crude oil and refined products markets. Quant Finance 12(12):1857–1875

22. de Brito RF, Oliveira AL (2014) Sliding window-based analysis of multiple foreign exchange trading systems by using soft computing techniques. In: 2014 International Joint Conference on Neural Networks (IJCNN). IEEE, pp 4251–4258

23. Deng J, Schuller B, Eyben F, Schuller D, Zhang Z, Francois H, Oh E (2020) Exploiting time-frequency patterns with LSTM-RNNS for low-bitrate audio restoration. Neural Comput Appl 32(4):1095–1107

24. Deng Y, Bao F, Kong Y, Ren Z, Dai Q (2016) Deep direct reinforcement learning for financial signal representation and trading. IEEE Trans Neural Netw Learn Syst 28(3):653–664

25. Dickey DA, Fuller WA (1979) Distribution of the estimators for autoregressive time series with a unit root. J Am Stat Assoc 74(366a):427–431

26. Ding S, Qu S, Xi Y, Wan S (2019) A long video caption generation algorithm for big video data retrieval. Fut Gener Comput Syst 93:583–595

27. Do B, Faff R (2010) Does simple pairs trading still work? Financ Anal J 66(4):83–95

28. Do B, Faff R, Hamza K (2006) A new approach to modeling and estimation for pairs trading. In: Proceedings of 2006 Financial Management Association European Conference. Citeseer, pp 87–99

29. Engle RF, Granger CW (1987) Co-integration and error correction: representation, estimation, and testing. Econom J Econom Soc, pp 251–276

30. Fallahpour S, Hakimian H, Taheri K, Ramezanifar E (2016) Pairs trading strategy optimization using the reinforcement learning method: a cointegration approach. Soft Comput 20(12):5051–5066

31. Ferrah A, Bradley K, Asher G (1992) Sensorless speed detection of inverter fed induction motors using rotor slot harmonics and fast Fourier transform. In: PESC'92 Record. In: 23rd Annual IEEE Power Electronics Specialists Conference. IEEE, pp 279–286

32. Fors A, Markiewicz O (2016) Swedish equities: casanovas or committed cointegrated partners

33. Gao L, Guo Z, Zhang H, Xu X, Shen HT (2017) Video captioning with attention-based LSTM and semantic consistency. IEEE Trans Multimed 19(9):2045–2055

34. Gatarek LT, Hoogerheide LF, van Dijk HK (2014) Return and risk of pairs trading using a simulation-based Bayesian procedure for predicting stable ratios of stock prices

35. Gatev E, Goetzmann WN, Rouwenhorst KG (2006) Pairs trading: performance of a relative-value arbitrage rule. Rev Financ Stud 19(3):797–827

36. Grenander U, Rosenblatt M, Blachman N (1957) Statistical analysis of stationary time series. PhT 10(6):47

37. Gutierrez JA, Tse Y (2011) Illuminating the profitability of pairs trading: a test of the relative pricing efficiency of markets for water utility stocks. J Trad 6(2):50–64

38. Hamaker EL, Dolan CV, Molenaar PC (2005) Statistical modeling of the individual: rationale and application of multivariate stationary time series analysis. Multivar Behav Res 40(2):207–233

39. Hansen BE (2001) The new econometrics of structural change: dating breaks in us labour productivity. J Econ Perspect 15(4):117–128

40. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780

41. Huang C, Min G, Wu Y, Ying Y, Pei K, Xiang Z (2017) Time series anomaly detection for trustworthy services in cloud computing systems. IEEE Trans Big Data 6:66

42. Huang SH, Shih WY, Lu JY, Chang HH, Chu CH, Wang JZ, Huang JL, Dai TS (2020) Online structural break detection for pairs trading using wavelet transform and hybrid deep learning model. In: 2020 IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, pp 209–216

43. Huang Z, Martin F (2019) Pairs trading strategies in a cointegration framework: back-tested on CFD and optimized by profit factor. Appl Econ 51(22):2436–2452

44. Ibrahim M, Torki M, ElNainay M (2018) Cnn based indoor localization using RSS time-series. In: 2018 IEEE Symposium on Computers and Communications (ISCC). IEEE, pp 1044–1049

45. Jiang YC, Cheam XJ, Chen CY, Kuo SY, Chou YH (2018) A novel portfolio optimization with short selling using gnqts and trend ratio. In: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, pp 1564–1569

46. Kang Y (2012) Real-time change detection in time series based on growing feature quantization. In: The 2012 International Joint Conference on Neural Networks (IJCNN). IEEE, pp 1–6

47. Karim F, Majumdar S, Darabi H, Chen S (2017) LSTM fully convolutional networks for time series classification. IEEE Access 6:1662–1669

48. Kim CJ, Morley JC, Nelson CR (2005) The structural break in the equity premium. J Bus Econ Stat 23(2):181–191

49. Kim JH (2014) Testing for parameter restrictions in a stationary var model: a bootstrap alternative. Econ Model 41:267–273

50. Kim T, Kim HY (2019) Optimizing the pairs-trading strategy using deep reinforcement learning with trading and stop-loss boundaries. Complexity 6:66

51. Kovács G, Sebestyen G, Hangan A (2019) Evaluation metrics for anomaly detection algorithms in time-series. Acta Universitatis Sapientiae Informatica 11:113–130

52. Krauss C (2017) Statistical arbitrage pairs trading strategies: review and outlook. J Econ Surv 31(2):513–545

53. Kuo SY, Cheam XJ, Jiang YC, Lai YT, Chang KN, Chou YH (2019) Portfolio optimization model using anqts with trend ratio on quadratic regression. In: 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC). IEEE, pp 629–634

54. Laptev N, Amizadeh S, Flint I (2015) Generic and scalable framework for automated time-series anomaly detection. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp 1939–1947

55. Lei K, Zhang B, Li Y, Yang M, Shen Y (2020) Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading. Expert Syst Appl 140:112872

56. Lei Y, Xu J (2015) Costly arbitrage through pairs trading. J Econ Dyn Control 56:1–19

57. Li Y, Zheng W, Zheng Z (2019) Deep robust reinforcement learning for practical algorithmic trading. IEEE Access 7:108014–108022

58. Liu J, Timmermann A (2013) Optimal convergence trade strategies. Rev Financ Stud 26(4):1048–1086
59. Liu P, Zhang H, Zhang K, Lin L, Zuo W (2018) Multi-level wavelet-CNN for image restoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp 773–782
60. Livieris IE, Pintelas E, Pintelas P (2020) A CNN-LSTM model for gold price time-series forecasting. Neural Comput Appl 66:1–10
61. Mitchell D, Białkowski J, Tompaidis S (2020) Volume-weighted average price tracking: a theoretical and empirical study. IISE Trans 52(8):864–889
62. Nourikhah H, Akbari MK, Kalantari M (2015) Modeling and predicting measured response time of cloud-based web services using long-memory time series. J Supercomput 71(2):673–696
63. Pendharkar PC, Cusatis P (2018) Trading financial indices with reinforcement learning agents. Expert Syst Appl 103:1–13
64. Perlin MS (2009) Evaluation of pairs-trading strategy at the Brazilian financial market. J Derivat Hedge Funds 15(2):122–136
65. Qiao W, Tian W, Tian Y, Yang Q, Wang Y, Zhang J (2019) The forecasting of pm2. 5 using a hybrid model based on wavelet transform and an improved deep learning algorithm. IEEE Access 7:142814–142825
66. Rad H, Low RKY, Faff R (2016) The profitability of pairs trading strategies: distance, cointegration and copula methods. Quant Finance 16(10):1541–1558
67. Radenović F, Tolias G, Chum O (2018) Fine-tuning CNN image retrieval with no human annotation. IEEE Trans Pattern Anal Mach Intell 41(7):1655–1668
68. Rioul O, Duhamel P (1992) Fast algorithms for discrete and continuous wavelet transforms. IEEE Trans Inf Theory 38(2):569–586
69. Sandryhaila A, Moura JM (2013) Discrete signal processing on graphs: graph Fourier transform. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, pp 6167–6170
70. Sarmento SM, Horta N (2020) Enhancing a pairs trading strategy with the application of machine learning. Expert Syst Appl 66:113490
71. Sarmento SM, Horta N (2020) Enhancing a pairs trading strategy with the application of machine learning. Expert Syst Appl 158:113490. https://doi.org/10.1016/j.eswa.2020.113490
72. Sarwar G (2012) Is vix an investor fear gauge in bric equity markets? J Multinatl Financ Manag 22(3):55–65
73. Selvin S, Vinayakumar R, Gopalakrishnan E, Menon VK, Soman K (2017) Stock price prediction using LSTM, RNN and CNN-sliding window model. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, pp 1643–1647
74. Siami-Namini S, Tavakoli N, Namin AS (2018) A comparison of arima and LSTM in forecasting time series. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, pp 1394–1401
75. Stoean C, Paja W, Stoean R, Sandita A (2019) Deep architectures for long-term stock price prediction with a heuristic-based strategy for trading simulations. PLoS ONE 14(10):e0223593
76. Su YJ, Hu WC, Jiang JH, Su RY (2020) A novel lmaeb-CNN model for Chinese microblog sentiment analysis. J Supercomput 66:1–15
77. Sunder S (1980) Stationarity of market risk: random coefficients tests for individual stocks. J Finance 35(4):883–896
78. Tao R, Deng B, Wang Y (2006) Research progress of the fractional Fourier transform in signal processing. Sci China Ser F 49(1):1–25
79. Vuagnoux M, Pasini S (2010) An improved technique to discover compromising electromagnetic emanations. In: 2010 IEEE International Symposium on Electromagnetic Compatibility. IEEE, pp 121–126
80. Wang S, Jiang J (2015) Learning natural language inference with LSTM. arXiv preprint arXiv:1512.08849
81. Yang J, Li J, Liu S (2020) A new algorithm of stock data mining in internet of multimedia things. J Supercomput 76(4):2374–2389
82. Zhang C, Wang Y, Chen C, Du C, Yin H, Wang H (2018) Stockassistant: a stock ai assistant for reliability modeling of stock comments. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 2710–2719

83. Zhang J, Wei Z, Yan Z, Zhou M, Pani A (2017) Online change-point detection in sparse time series with application to online advertising. IEEE Trans Syst Man Cybernet Syst 49(6):1141–1151

84. Zhang K, Zuo W, Chen Y, Meng D, Zhang L (2017) Beyond a gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans Image Process 26(7):3142–3155

85. Zhang L, Aggarwal C, Qi GJ (2017) Stock price prediction via discovering multi-frequency trading patterns. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp 2141–2149

86. Zhang Z, Zohren S, Roberts S (2019) Deeplob: deep convolutional neural networks for limit order books. IEEE Trans Signal Process 67(11):3001–3012

87. Zhao Y, Shen Y, Zhu Y, Yao J (2018) Forecasting wavelet transformed time series with attentive neural networks. In: IEEE International Conference on Data Mining (ICDM). IEEE, pp 1452–1457

88. Zhou F, Hm Zhou, Yang Z, Yang L (2019) Emd2fnn: a strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction. Expert Syst Appl 115:136–151