Detecting Market Manipulation with Self-supervised Learning: A Hybrid Framework Using Frequency-based Synthetic Anomaly and Domain-Specific Features

Yongsheng Dai, Barry Quinn, Fearghal Kearney, Weilong Liu, Ivor Spence, Karen Rafferty, and Hui Wang, *Senior Member, IEEE*

***Abstract*—Market manipulation significantly undermines market integrity and investor confidence in financial systems. Despite increasing research attention, existing detection methods often fall short of practical deployment requirements due to signal concealment by sophisticated manipulators, scarcity of labelled training data, and difficulty in precisely localising manipulation boundaries. This paper introduces SDFM, a Self-supervised Detection Framework tailored for financial Market manipulation that addresses these fundamental challenges through three innovative components. First, our Amplification Component extracts and fuses domain-specific features grounded in market microstructure theory, substantially amplifying subtle manipulation signals that would otherwise remain concealed. Second, our Synthesis Component generates realistic synthetic anomalies through few-shot learning and dynamic frequency analysis using Discrete Wavelet Transform, enabling self-supervised training without relying on scarce labelled data. Third, our Detection Component employs a novel Dual-branch Contrastive Detection network that enhances sensitivity to manipulation boundaries through local contrastive learning. Evaluation on a unique dataset of 25 Chinese stock market manipulation cases demonstrates that SDFM consistently outperforms state-of-the-art methods across 11 metrics, achieving 77.59% precision and 5.44 mean detection delay, representing substantial improvements for practical financial surveillance applications.**

***Index Terms*—Market Manipulation, self-supervised learning, time series anomaly detection, frequency-based synthetic anomaly, domain-specific features.**

I. Introduction

Market manipulation activities, such as insider trading and spoofing, undermine financial market integrity and stability, reduce investor confidence, and erode the efficiency of price discovery. However, such manipulations are difficult to detect, with the dynamic and opaque nature of modern trading venues making identification difficult, see for example [1]–[3]. This search is further compounded by the lack of precisely labelled datasets [4], [5]. These challenges motivate the need for detection frameworks that can operate under imperfect conditions, such as datasets without original annotations.

Theoretical insights suggest that market manipulation practices lead to abnormalities in financial transaction data [6]. Therefore, an increasing number of studies seek to detect market manipulation using transaction data such as stock market trades. This approach leads to a type of time series anomaly detection task. However, the outcomes of these detection processes often fall short of expectations and remain insufficient for practical deployment, due to the following challenges:

1) **Signal Concealment:** Existing methods typically achieve manipulation detection directly based on the original features of the transaction data, such as price and volume [7], [8]. However, the manipulators' sophisticated fraud techniques are meticulously designed to prevent their actions from leaving conspicuous traces amongst the large number of normal transactions. This is often achieved by strategically mixing truthful and false information [9], [10]. As a result, only very subtle anomaly patterns remain, making it difficult to detect manipulations.
2) **Data Sparsity:** Increasingly advanced big data techniques have enabled corresponding AI methods to become dominant across many fields. However, it remains challenging to implement a successful data-driven algorithm oriented for financial manipulation detection, due to the scarcity of anomaly annotation and training samples [11]. To tackle this issue, some prior detection algorithms often adopt an unsupervised strategy. These typically rely on the reconstruction error of learned normal samples to highlight anomalies. However, such strategies can suffer from high false alarm rates as they struggle to learn the unique patterns of anomalies [12], [13].
3) **Boundary Vagueness:** In addition to successfully detecting the presence of manipulation (as noted in the first challenge), modern detection frameworks also seek to precisely locate the manipulation interval. However, due to rapid market dynamics and fine-grained point-level resolution, anomalous intervals in transaction data are often easily confounded with their adjacent normal intervals [14]. Furthermore, literature has demonstrated that manipulators often engage in multiple rounds of

This work is supported by the PwC Research and Development Centre (R5212ECS) and the Multimodal Video Search by Examples (MVSE) project funded by UK EPSRC (EP/V002740/2). *(Corresponding author: Hui Wang.)*

Yongsheng Dai, Ivor Spence, Karen Rafferty, and Hui Wang are with the School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast, Northern Ireland (e-mail: ydai09@qub.ac.uk; I.Spence@qub.ac.uk; K.Rafferty@qub.ac.uk; h.wang@qub.ac.uk).

Barry Quinn is with Ulster University Business School, Ulster University, Belfast, Northern Ireland (e-mail: b.quinn1@ulster.ac.uk).

Fearghal Kearney is with Queen's Business School, Queen's University Belfast, Belfast, Northern Ireland (e-mail: f.kearney@qub.ac.uk).

Weilong Liu is with Lingnan College, Sun Yat-sen University, Guangzhou, China (e-mail: liuwlong7@mail.sysu.edu.cn).

trading [15], [16]. Consequently, precisely locating manipulation boundaries remains a very difficult task, with delays in detecting these boundaries leading to substantial economic losses [6].

In this paper, we introduce a **S**elf-supervised **D**etection **F**ramework tailored to financial market **M**anipulation, named **SDFM**, to detect and locate market manipulations within our stock market transaction data. To address the challenges outlined above, our SDFM takes the following steps: (1) To deal with the first challenge, SDFM incorporates domain-specific features throughout the self-supervised learning and detection processes. More specifically, it incorporates behavioural anomalies that have appeared in manipulation cases, namely, temporal clustering of orders and volatility in order flow intensity. This substantially amplifies the anomaly signals, making it much easier to identify the presence of various types of manipulation. (2) To deal with the second challenge, the training of the detection network within SDFM does not rely on original labeled samples, but rather on self-supervision supported by our realistic synthetic anomalies. These anomalies come from a novel generation component based on the few-shot learning of domain-knowledge and the dynamic frequency analysis of stock market transaction streams. (3) To deal with the third challenge, SDFM uses our Dual-branch Contrastive Detection (DCD) network. It significantly enhances SDFM's sensitivity to anomaly boundaries and markedly improves its ability to discriminate abnormal intervals from adjacent normal sequences.

Specifically, SDFM realizes these solutions mainly through three components: First, solution (1) above is supported by the **Amplification Component**. It exploits diverse domain-knowledge to extract multiple domain-specific features from conventional stock market transaction data. They are then fused at different granularities, serving as the foundation for all subsequent processes within our SDFM. This fusion results in a comprehensive expression and a thorough exploration of manipulation patterns. In addition, the process of extracting these features leads to the refinement and clustering of unique traces of anomalous transaction behaviors from multiple perspectives, and their integration allows representative anomaly signals to be repeatedly amplified. In financial markets, there is a need to amplify subtle anomaly signals that manipulators attempt to conceal within normal market activity [3]. Thus, compared to the original transaction data, the new temporal representation during manipulation intervals possesses a more obvious intensity and becomes more distinguishable from normal intervals. This enhancement makes it much easier to identify the presence of various types of market manipulation. Second, our solution (2) is supported by the **Synthesis Component**. In this process, we introduce an innovative synthetic anomaly generation method, that begins with a few-shot learning of the domain knowledge provided by the previous component. This leads to the synthetic data reflecting real-world manipulation patterns, as well as allowing it to inherit the amplification effect of anomaly signals. Both the few-shot learning and anomaly synthesizing above are based on the frequency distribution of the time series, meaning that changes of frequency information play a significant role in many practical scenarios, including financial markets [17], [18]. Therefore, SDFM applies the Discrete Wavelet Transform (DWT) [19] here to dynamically perform multi-scale representation of frequency at different resolutions and time points, which is particularly beneficial for analyzing signals that exhibit non-stationary behavior, such as stock market data [14]. In addition, this transformation process can capture and emphasize important features that are robust against variations in time series, thus also providing a further refinement of anomaly signals. Third, our solution (3) is supported by the **Detection Component**. During training, the DCD network employs self-supervised learning to understand manipulation patterns, aided by the final component. Subsequently, in the testing phase, this detection component conducts temporal alignment to the outputs of the DCD network, leading to the timely and accurate identification of manipulation intervals within the original stock market trade data. The contrast-based DCD framework, with its dual self-supervised learning branches, has the ability to detect anomalies from a variety of perspectives. This comprehensive approach ensures that it can identify the unique patterns in market manipulation datasets. Additionally, the DCD network incorporates a novel contrastive pretext task based on local sub-segments, which enhances the network's discrimination between anomalous and adjacent normal intervals. This improvement significantly boosts the network's sensitivity to the anomaly boundaries and its responsiveness across the entire anomaly range. Consequently, our approach can detect the occurrence of manipulations more quickly and can ascertain their duration more accurately.

In general, the contributions of this study can be summarized as follows:

- We propose a novel self-supervised detection framework for market manipulation named SDFM, where a Dual-branch Contrastive Detection (DCD) network can characterize manipulation from different perspectives, ensuring that the unique patterns of these anomalies are deeply acquired. Additionally, an innovative contrastive pretext task within DCD significantly boosts the network's sensitivity to the anomaly boundaries and its responsiveness across the entire anomaly range. Consequently, our approach can detect the occurrence of manipulation more quickly and can localize their duration more accurately.
- We propose a new synthetic anomaly generation method based on the few-shot learning of domain-knowledge and the dynamic frequency analysis of market manipulation. Given the scarcity of annotated data in practice, our approach enables the generation of unlimited anomaly samples using the natural simulation of real-world manipulation patterns, supporting the self-supervised learning of our detection framework. To the best of our knowledge, this is the first attempt to completely utilize the dynamic frequency distribution of time series as both the data perturbation target and the synthetic outcome in an artificial anomaly generation method.
- We propose a novel method that leverages domain-

specific features to facilitate the detection of market manipulations. These features, derived from various domains, provide a comprehensive expression and thorough exploration of manipulation patterns. Innovatively, we set their fusion results as the foundation for all subsequent self-supervised learning and detection processes within our framework. Our approach refines and clusters unique traces of anomalous transaction behaviors from multiple perspectives, which enables the anomaly signals to be amplified repeatedly. Consequently, the temporal representation during manipulation intervals exhibits substantially increased intensity and distinctiveness, making it much easier to identify the presence of various types of manipulation.

- We evaluate our approach using a real-world case study, based on a unique dataset of Chinese stock market manipulation cases. Our framework achieves performance that is consistently superior to 10 state-of-the-art baseline methods across a comprehensive assessment of 11 metrics. Extensive quantitative and qualitative experiments demonstrate that our framework can generate a much faster and stronger response to real-world manipulation, offering new insights into financial market manipulation and establishing a powerful framework that can be used to detect these anomalies.

II. Related Work

A. Market Manipulation

Market manipulation encompasses a range of strategies designed to create artificial prices or misleading market conditions. The theoretical literature establishes three primary categories: **information-based manipulation**, where false information is disseminated to influence prices; **trade-based manipulation**, involving coordinated trading patterns to create artificial demand or supply; and **action-based manipulation**, where manipulators' actions create false market signals [3].

Vila (1989) pioneered theoretical analysis of information-based manipulation, demonstrating how manipulators can profit from releasing false information whilst maintaining credibility [1]. Subsequent work by Benabou and Laroque (1992) and Van Bommel (2003) addressed credibility concerns by incorporating imprecise information and rumour-spreading mechanisms, showing that manipulators can maintain long-term profitability by strategically mixing truthful and false information [9], [10].

Trade-based manipulation has received extensive theoretical attention. Kumar and Seppi (1992) modelled closing price manipulation in futures markets [20], whilst Hillion and Suominen (2004) demonstrated how market design features such as closing call auctions can reduce manipulation opportunities [21]. Empirical studies have documented various manipulation strategies in practice, with Aggarwal and Wu (2006) analyzing 142 pump-and-dump cases and finding significant heterogeneity in manipulation duration and methods [2].

Detection Methodologies: The detection challenge is compounded by manipulators' sophisticated concealment techniques and the scarcity of labelled training data. Traditional approaches have evolved from supervised learning frameworks using decision trees [22] and support vector machines [23] to advanced machine learning methodologies. James, Leung, and Prokhorov (2023) demonstrate significant advances using dynamic time warping combined with extreme value theory, achieving 90% detection rates compared to 60% for traditional econometric models [11]. Their one-class machine learning approach addresses the practical limitation that brokers often lack access to comprehensive databases of historical illegal trading activity.

Network-based approaches have emerged as promising alternatives, with Sun et al. (2017) proposing multi-slice trading network analysis to identify anomalous traders through temporal relationship patterns [24]. These methods complement traditional statistical approaches by capturing the collaborative nature of manipulation strategies, particularly relevant in emerging markets where coordinated trading groups are prevalent.

The temporal dimension has proven crucial, with literature demonstrating that informed traders engage in multiple rounds of trading based on private information [15], [16]. This insight has driven development of time-series-specific detection models that explicitly account for sequential trading behaviour rather than treating individual transactions independently.

Recent research has examined how legal enforcement affects manipulative behaviour. Kacperczyk and Pagnotta (2024) demonstrate that prosecution risk significantly influences insiders' strategic behaviour, with manipulators adjusting the aggressiveness and timing of their trades to minimise detection probability [25]. This behavioural adaptation underscores the importance of developing sophisticated detection systems that can identify evolving manipulation strategies whilst accounting for manipulators' strategic responses to regulatory oversight.

B. Learning Schemes

Training deep learning networks for anomaly detection involves four learning schemes: supervised, semi-supervised, unsupervised, and self-supervised, depending on the availability of annotations during training.

Supervised methods aim to learn class boundaries based on all labels in the training set but are not applicable to time series anomaly detection due to the unknown or improperly labeled nature of anomalies [26]. Semi-supervised methods train models based on a context where the dataset consists only of normal points, detecting deviations from this distribution as anomalies. However, these methods have limitations, as discussed earlier.

Unsupervised learning is flexible but faces challenges in accurately detecting anomalies without supervision and may suffer from subject and environment-dependency and evaluation difficulties [27]–[29]. By leveraging generative models trained by reconstruction errors, these unsupervised methods identify anomalies by treating any deviations from learned normal patterns as outliers. However, given the drastic changes in real-world temporal contexts and the difficulty in accounting for all normal patterns during training, these unsupervised

methods are prone to high false alarm rates [12], [13]. Additionally, real-world anomalies are not merely simple outliers or extreme values, similar temporal patterns may represent different categories in different contexts [30]. However, unsupervised learning methods fail to include such information for model training, which can further inhibit the performance of discovering specific patterns associated with anomalies.

Self-supervised learning, like unsupervised learning, does not rely on original annotations and tends to have more stable performance. However, it requires designing appropriate pseudo labels and pretext tasks [31]. In this paper, we create these components based on the synthetic anomalies from our Synthesis Component, which will be introduced in detail in Section 3.4.

C. Contrastive Learning for Time Series

Contrastive learning is a widely used self-supervised learning strategy. The goal is to learn an encoder that maps inputs into an embedding space in which similar data samples (positive) are close to each other while dissimilar (negative) ones are far apart. It can help the network to distinguish any instance from the others more sensitively and accurately. Contrast learning was initially investigated mainly in computer vision tasks [32]. But it is also increasingly being applied to various time series problems. For instance, to enhance model generalization capacity in time series segmentation tasks, Xiao et al. [29] explored unlabelled target data using contrastive learning to enable the model to capture its characteristics. When building a time series pre-training model, Zhang et al. [33] simultaneously mined time domain information and frequency domain information based on contrastive learning. Eldele et al. [31] proposed a representation learning framework for time series classification task via temporal and contextual contrastive learning. Eldele et al. [34] employed a dual attention contrastive representation learning network in time series anomaly detection.

In these previous research works, people modeled the entire time series sample globally, and contrastive learning was only applied to two complete long sequences from different sources or processes. However, in time series anomaly detection, anomalies are sharp changes in data distribution, which are always rare and only occur in relatively short segments. Applying the above methods directly to this task does not take advantage of contrastive learning to fully explore the enormous differences between abnormal segments and their adjacent normal segments in the same long-term sequence. As a result, the network's sensitivity to the boundaries between these two types of fragments in the local area will not be sufficiently trained and improved. This will affect the network's ability to detect the occurrence of anomalies in a timely manner, which is critical for many application scenarios such as IoT and finance. This is also the main motivation for us to use contrastive learning to locally model the continuous subsegments of the abnormal area in the same time series sample in our DCD network.

III. Methodology

A. Overview

The overall structure of our SDFM is summarized in Fig. 1. It mainly consists of three components: First, **Amplification Component**. This component extracts and fuses multiple domain-specific features from stock transaction data, enhancing the intensity and distinguishability of manipulation signals. By refining and clustering unique traces of anomalous behaviors, it lays a robust foundation for subsequent analysis, allowing for clearer discrimination between normal and manipulative intervals. Second, **Synthesis Component**. Leveraging few-shot learning informed by domain knowledge, this component constructs a novel synthetic anomaly generation method that mimics real-world manipulation patterns. Utilizing Discrete Wavelet Transform (DWT), it dynamically analyzes the frequency distribution of a financial transaction stream, facilitating multi-scale representation and enhancing the robustness of anomaly signals against time variations. Third, **Detection Component**. Incorporating a Dual Contrastive Detection (DCD) network, this component performs self-supervised learning to recognize manipulation patterns. During testing, it aligns the outputs of the DCD network to timely and accurately localize manipulation intervals within the original transaction data. In the DCD network, a novel contrastive pretext task based on local sub-segments significantly improve the framework's discrimination between the anomalous interval and adjacent normal sequences, enhancing sensitivity to anomaly boundaries and overall responsiveness.

B. Amplification Component

This component extracts multiple domain-specific features from conventional stock transaction data, each based on distinct domain knowledge. These features are then fused at varying granularities, forming the foundation for all subsequent processes within the SDFM framework. This approach not only enables a comprehensive representation and an in-depth exploration of manipulation patterns but also refines and clusters the unique traces of anomalous transaction behaviors from diverse perspectives. As a result, the temporal representation during manipulation intervals exhibit substantially increased intensity and distinctiveness, making it much easier to identify the presence of various types of manipulation.

1) Domain-specific Features Extractor: Drawing from established market microstructure literature, we extract features that capture manipulative trading patterns documented in academic research. Our approach leverages two key behavioural anomalies identified in manipulation cases: temporal clustering of orders and volatility in order flow intensity.

Rush Order (RO) Feature: Based on empirical evidence that manipulators often execute rapid sequential orders to create artificial price movements [2], we define rush orders as multiple consecutive transactions of identical order type (buy or sell) executed at the same timestamp. This feature captures the "clustering" behaviour documented in pump-and-dump schemes, where coordinated trading creates artificial market activity. Mathematically, for each timestamp t, we identify

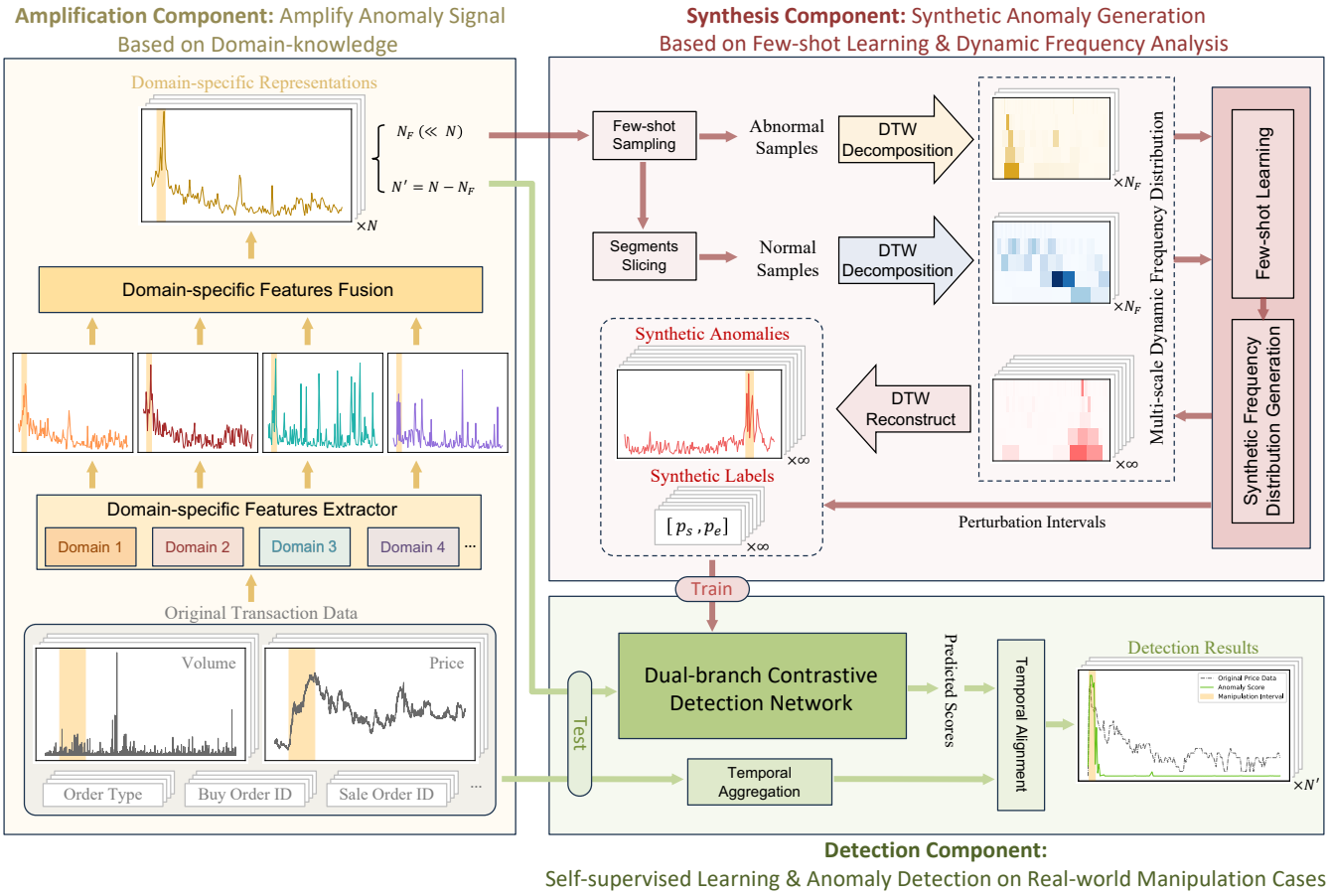

Fig. 1. The overall architecture of our SDFM framework.

rush orders when $count_{B/S}(t) > 1$, where $count_{B/S}(t)$ represents the number of buy or sell orders at time t.

The theoretical foundation for this feature derives from trade-based manipulation literature, which demonstrates that manipulators must execute trades in clusters to achieve meaningful price impact whilst minimising market impact costs [35]. This clustering behaviour contradicts the random arrival patterns expected under normal trading conditions.

Order Cancellation Ratio (OCR) Feature: Based on extensive literature documenting that abnormally high order cancellation rates are indicative of manipulative strategies such as spoofing, layering, and quote stuffing [36], we compute OCR to capture rapid order placement and cancellation patterns. The US Securities and Exchange Commission reports that 96.8% of orders are typically cancelled before execution, with 90% cancelled within one second under normal conditions [36].

OCR quantifies the intensity of order book manipulation by measuring the ratio of cancelled orders to total submitted orders within each time window. High OCR values beyond normal market-making levels may indicate spoofing behaviour, where manipulators submit large orders to create false impressions of market demand or supply, only to cancel these orders before execution [37].

Specifically, we calculate: $OCR_t = \frac{\text{Cancelled Orders}_t}{\text{Total Submitted Orders}_t}$, where abnormally high values suggest potential market manipulation attempts. This feature is particularly effective at detecting quote stuffing and layering strategies, where manipulators flood the market with orders they intend to cancel [36]. The calculation procedures for OCR_B and OCR_S are detailed in Algorithm 1.

These features address the fundamental challenge identified in manipulation detection literature: the need to amplify subtle anomaly signals that manipulators attempt to conceal within normal market activity [3].

2) Domain-specific Features Fusion: The left part of Fig. 1 has initially showcase the fusion result of one real-world manipulation case. In order to demonstrate the amplification effectiveness more apparently and visualize each extracted domain-specific feature with more evident details, Fig. 2 illustrates this fusion process using another example. As we can see, these features can explore and cluster the manipulation patterns from multiple perspectives, and their fusion allows the representation of anomaly signals to be amplified repeatedly. Compared to the original transaction data (we use original volume and price data as examples at the bottom of Fig. 2), the newly derived temporal representation during manipulation intervals exhibits markedly increased intensity and distinguishability from normal intervals.

Leveraging a temporal merging process, we first aligns and combines data from different features by synchronizing entries

Algorithm 1 Order Cancellation Ratio (OCR) Feature Extraction

Require: Raw trading data D_{raw} with: Time, OrderID, OrderType (B/S), OrderStatus (Submitted/Cancelled/Executed), Volume
Ensure: OCR features for buy orders OCR_B and sell orders OCR_S
for each timestamp and order type combination **do**
  $Orders^B_{submitted}(t) \leftarrow$ Count of buy orders submitted at time t
  $Orders^B_{cancelled}(t) \leftarrow$ Count of buy orders cancelled at time t
  $Orders^S_{submitted}(t) \leftarrow$ Count of sell orders submitted at time t
  $Orders^S_{cancelled}(t) \leftarrow$ Count of sell orders cancelled at time t
end for
for each timestamp **do**
  if $Orders^B_{submitted}(t) > 0$ **then**
    $OCR_B(t) \leftarrow \frac{Orders^B_{cancelled}(t)}{Orders^B_{submitted}(t)}$
  else
    $OCR_B(t) \leftarrow 0$
  end if
  if $Orders^S_{submitted}(t) > 0$ **then**
    $OCR_S(t) \leftarrow \frac{Orders^S_{cancelled}(t)}{Orders^S_{submitted}(t)}$
  else
    $OCR_S(t) \leftarrow 0$
  end if
end for
for each time window $W_k = [t_k, t_k + f)$, $k = 1, 2, \ldots$ **do**
  ▷ Temporal aggregation with frequency f
  $OCR^{agg}_B(W_k) \leftarrow \text{mean}_{t \in W_k} OCR_B(t)$
  $OCR^{agg}_S(W_k) \leftarrow \text{mean}_{t \in W_k} OCR_S(t)$
end for
return $OCR_B = \{OCR^{agg}_B(W_k)\}^K_{k=1}, OCR_S = \{OCR^{agg}_S(W_k)\}^K_{k=1}$

Algorithm 2 Rush Order (RO) Feature Extraction

Require: Raw trading data D_{raw} with: Time, OrderType (B/S), Price
Ensure: RO features for buy orders RO_B and sell orders RO_S
Filter data for buy orders: $D_B \leftarrow \{d \in D_{raw} : d.OrderType = \text{B}\}$
Filter data for sell orders: $D_S \leftarrow \{d \in D_{raw} : d.OrderType = \text{S}\}$
for each timestamp t in $D_B \cup D_S$ **do** ▷ Analyze transaction clustering
  $count_B(t) \leftarrow |\{d \in D_B : d.Time = t\}|$ ▷ Count buy orders at t
  $count_S(t) \leftarrow |\{d \in D_S : d.Time = t\}|$ ▷ Count sell orders at t
end for
for each timestamp t **do** ▷ Identify manipulation clusters
  if $count_B(t) > 1$ **then**
    $rush\_ind_B(t) \leftarrow 1$ ▷ Buy-side clustering indicates coordination
  else
    $rush\_ind_B(t) \leftarrow 0$ ▷ Normal single transaction
  end if
  if $count_S(t) > 1$ **then**
    $rush\_ind_S(t) \leftarrow 1$ ▷ Sell-side clustering indicates coordination
  else
    $rush\_ind_S(t) \leftarrow 0$ ▷ Normal single transaction
  end if
end for
for each time window $W_k = [t_k, t_k + f)$, $k = 1, 2, \ldots$ **do**
  ▷ Temporal aggregation with frequency f
  $RO^{agg}_B(W_k) \leftarrow \sum_{t \in W_k} rush\_ind_B(t)$ ▷ Sum buy rush orders in window
  $RO^{agg}_S(W_k) \leftarrow \sum_{t \in W_k} rush\_ind_S(t)$ ▷ Sum sell rush orders in window
end for
return $RO_B = \{RO^{agg}_B(W_k)\}^K_{k=1}, RO_S = \{RO^{agg}_S(W_k)\}^K_{k=1}$

across a common time axis. An inner join is employed to ensure that only those records with a corresponding time stamp in both datasets are combined. As a result, we will obtain a merging data sheet with unified time index. We use RO_B^t, RO_S^t, OCR_B^t and OCR_S^t, to represent these features of the same time index in the merging data sheet. This method is particularly useful for financial time series data, where alignment on time is crucial for maintaining the sequence's integrity and ensuring that the analysis reflects true market behaviors without introducing temporal discrepancies. After that, we apply the following formulas in this fusion module to incorporate different features into a new feature F_{Domian}, which constitute the domain-specific representations.

$$\text{scale}\{OCR^t\} = \frac{OCR^t - OCR_{min}}{OCR_{max} - OCR_{min}} \times (r_{max} - r_{min}) + r_{min} \quad (1)$$

$$F^t_{Domian} = (RO^t_B * \text{scale}\{OCR^t_B\}) + (RO^t_S * \text{scale}\{OCR^t_S\}) \quad (2)$$

where OCR_{min} and OCR_{max} represent the minimum and maximum value in the original OCR feature set, r_{min} and r_{max} are two hyper-parameters, indicating the lower and upper bound of the target scaling range.

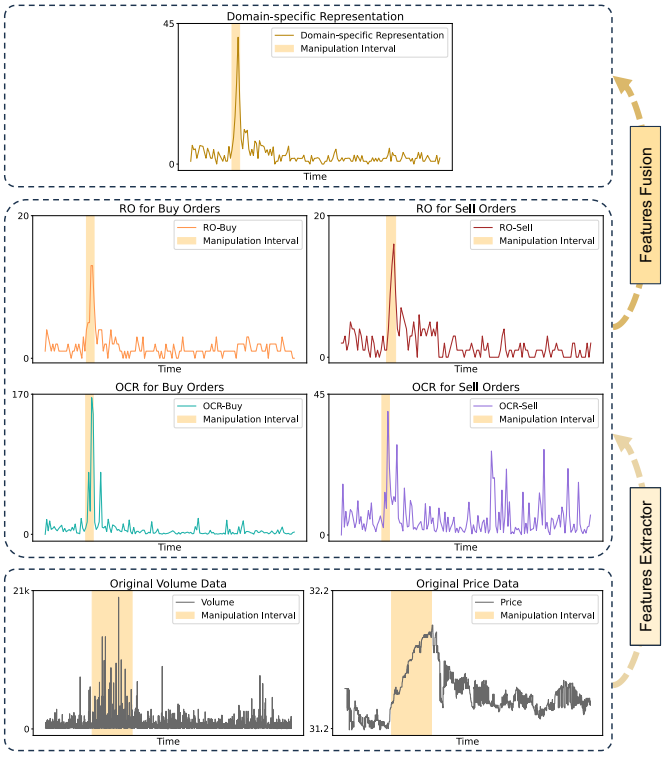

Fig. 2. The effectiveness of domain-specific features extracting and fusion.

C. Synthesis Component

1) Few-shot Learning on Real-world Manipulation Patterns: We introduce a new synthetic anomaly generation method in this component, which initiates with few-shot learning based on the domain knowledge provided by the preceding

component. This process not only inherits the amplification effect of anomaly signals but also ensures fidelity between the synthetic outcomes and real world manipulation patterns. Moreover, as changes in frequency information play a critical role in time series data across various practical scenarios, including financial markets [17], [18], both the initial few-shot learning and the subsequent anomaly synthesis are grounded in the frequency distribution of the time series.

Specifically, this synthesis component randomly draws a few abnormal examples from the domain-specific representations of the amplification component. After that, SDFM applies a dynamic time-frequency converter to decompose these time series into multi-scale time-varying frequency distributions, which will be described in detail in the next section. During the few-shot learning, we carefully observe the characteristics and regularities of the dynamic frequency distribution for real-world abnormal examples. These observation results will direct the subsequent processes of data perturbation and synthetic anomaly generation. Simultaneously, the normal segments within abnormal examples undergo the same processes of frequency decomposition and few-shot learning. They direct the construction of fundamental signals during subsequent data perturbation.

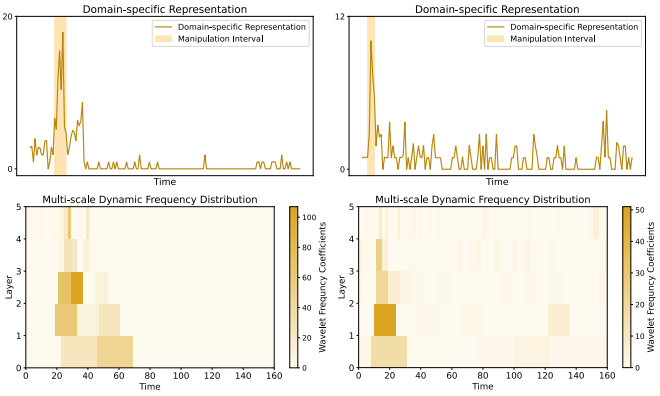

Fig. 3. The multi-scale dynamic frequency distribution of manipulation samples after DWT decomposition.

2) Dynamic Frequency Analysis based on DWT: To the best of our knowledge, this is the first attempt to fully utilize the dynamic frequency distribution of a time series as both the data perturbation target and the synthetic outcome in an artificial anomaly generation method. SDFM applies the Discrete Wavelet Transform (DWT) [19] here to dynamically perform multi-scale representation of frequency at different resolutions and time points, which is particularly beneficial for analyzing signals that exhibit non-stationary behavior, such as transaction data [14]. Compared to conventional time-frequency converters such as Fourier Transforms [38] that only generate static frequency representation for an entire sequence, DWT excels in providing simultaneous time and frequency information, enabling the analysis of how different frequency components of a signal vary over time.

In addition, DWT can efficiently represent signal discontinuities and sharp transitions, capturing significant features that are robust to variations in the signal. It often yields a sparse representation where most of the wavelet coefficients are zero or near-zero, and only a few significant coefficients carry most of the signal information [39]. Consequently, performing time series decomposition based on DWT also provides a further refinement of anomaly signals.

The DWT decomposes a discrete signal $x[n]$ by projecting it onto a family of wavelet basis functions. These functions are derived from a mother wavelet $\psi(t)$ through scaling and translation:

$$\psi_{j,k}(t) = 2^{j/2}\psi(2^j t - k) \tag{3}$$

where j and k are integers representing the scaling and translation parameters, respectively.

The DWT of a discrete signal $x[n]$ can be expressed as:

$$W_\psi[j,k] = \sum_{n=0}^{N-1} x[n]\psi_{j,k}[n] \tag{4}$$

where $W_\psi[j,k]$ represents the wavelet coefficient at scale j and position k.

In practice, DWT is efficiently implemented using a filter bank approach with high-pass ($g[n]$) and low-pass ($h[n]$) filters, followed by dyadic downsampling. Specifically, the signal $x[n]$ is convolved with both filters to extract approximations and details, respectively. After filtering, each output is down-sampled by 2 to reduce the sampling rate by half, which effectively doubles the frequency resolution at each subsequent scale. This is because each level of decomposition doubles the time scale:

$$y_{\text{high}}[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot g[n-k] \tag{5}$$

$$y_{\text{low}}[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \tag{6}$$

$$d_1[n] = y_{\text{high}}[2n] \tag{7}$$

$$a_1[n] = y_{\text{low}}[2n] \tag{8}$$

where $d_1[n]$ represents the detail coefficients (high-frequency components) and $a_1[n]$ represents the approximation coefficients (low-frequency components) at the first level.

After that, the DWT can be extended to multiple levels through the recursive decomposition of the approximation coefficients. This creates a hierarchical representation where Level 1 detail coefficients (d_1) capture the highest frequency components; Level 2 detail coefficients (d_2) capture the next highest frequency band; so on for levels $j = 3, 4, ..., J$; Level J approximation coefficients (a_J) capture the lowest frequency components:

$$a_{j+1}[n] = \sum_k h[k-2n]a_j[k] \tag{9}$$

$$d_{j+1}[n] = \sum_k g[k-2n]a_j[k]. \tag{10}$$

The original signal can be reconstructed using:

Algorithm 3 Synthetic Frequency Distribution Generation

Require: The frequency distribution of normal segment: $coef\_nor$, and its layer number: J; the max range of perturbation: p^{max}; the max and min intensities and bias of perturbations: $A^{max}, A^{min}, B^{max}, B^{min}$; the bias of perturbation position: p_r
Ensure: The sets of synthetic frequencies $Self\text{-}F$ and perturbation intervals $Self\text{-}L$

for $i : 1 \rightarrow N_s$ **do** ▷ N_s: the required number of synthetic samples
  Basic factors: $per', bia' \leftarrow \text{rand}(0, 1) \in \mathbb{R}^{\text{shape}(coef\_nor)}$
  New signal: $coef' \leftarrow coef\_nor \cdot per' + bia'$
  for $l : J - 1 \rightarrow 0$ **do** ▷ Perturb frequency coefficients in each layer
    if $l = J - 1$ **then**
      $p_s[l] \leftarrow \text{rand}(0, |coef'[l]| - 1)$
      $p_e[l] \leftarrow \min(p_s + \text{rand}(1, p^{max}), |coef'[l]|)$
    else
      $p_s[l] \leftarrow \min(0, (p_s[l + 1]//2) + \text{rand}(-p_r, p_r))$
      $p_e[l] \leftarrow \max(p_s[l] + 1, (p_e[l + 1]//2) + \text{rand}(-p_r, p_r))$
    end if
    $per[l] \leftarrow \{\text{rand}(A^{min}, A^{max})\}_1^{p_e[l]-p_s[l]}$
    $bia[l] \leftarrow \{\text{rand}(B^{min}, B^{max})\}_1^{p_e[l]-p_s[l]}$
    $coef'[l][p_s[l] : p_e[l]] \leftarrow coef'[l][p_s[l] : p_e[l]] \cdot per[l] + bia[l]$
  end for
  $Self\text{-}F$.add($coef'$) ▷ The basis for synthetic anomalies
  $Self\text{-}L$.add($[p_s[J - 1], p_e[J - 1]]$) ▷ The basis for synthetic labels
end for
return $Self\text{-}F$, $Self\text{-}L$

$$x[n] = \sum_k a_J[k]\phi_{J,k}[n] + \sum_{j=1}^{J}\sum_k d_j[k]\psi_{j,k}[n] \tag{11}$$

where $\phi_{j,k}[n]$ represents the scaling function. In multi-resolution analysis via the DWT, only retaining the approximation coefficients from the final level (a_J) is both theoretically sound and computationally efficient. They capture the most significant, overarching trends in the data. This selective retention helps in focusing the analysis on the macro-level features that influence anomaly detection, reducing the noise and complexity associated with lower-level, finer-grained details.

In addition, we further leverage Mallat algorithm [40] to formalize the above decomposition as:

$$a_{j+1}[p] = \sum_n h[n - 2p]a_j[n] \tag{12}$$

$$d_{j+1}[p] = \sum_n g[n - 2p]a_j[n] \tag{13}$$

with reconstruction:

$$a_j[n] = \sum_p \tilde{h}[n - 2p]a_{j+1}[p] + \sum_p \tilde{g}[n - 2p]d_{j+1}[p] \tag{14}$$

where $\tilde{h}$ and $\tilde{g}$ are the reconstruction filters.

This multi-resolution analysis enables the observation of financial time series across different frequency bands, facilitating the detection of anomalies that may manifest at specific scales, making DWT particularly valuable for financial time series anomaly detection.

Fig. 3 leverages two abnormal example to visualize the effectiveness of DWT decomposition. As can be seen in the final heat map, the energy distribution becomes concentrated within the anomaly region. At the higher level (i.e. the higher frequency component) the energy concentration becomes more pronounced. It demonstrates that the distribution of frequency energy in the DWT decomposition resembles a pyramid pattern. The discovery of this phenomenon will be relied upon to guide our subsequent anomaly synthesis process.

3) Synthetic Frequency Distribution Generation: Algorithm 3 above shows the process of synthetic frequency distribution generation. In this paper, we apply $db2$ [41] as the filter in our wavelet transform. This process finally generates the synthetic frequency $Self\text{-}F$ and logged perturbation intervals $Self\text{-}L$. The former will be reconstructed into time series $Self\text{-}Y$ (i.e., the synthetic anomalies) by DWT and serve as the pseudo training samples for subsequent self-supervised learning. The later will also serve as the pseudo labels for it.

D. Detection Component

During training, the Dual-branch Contrastive Detection (DCD) network in this component realizes the self-supervised learning for manipulation patterns with the support of synthetic samples from Synthesis Component. In the testing phase, this detection component performs temporal alignment on the outputs of the DCD network, thereby generating timely and accurate localization of manipulation intervals within the original transaction data. The DCD, a contrast-based detection network with dual self-supervised learning branches, excels at characterizing transaction anomalies from various perspectives. This capability facilitates comprehensive modeling of market manipulation, ensuring in-depth learning of their unique anomaly patterns. Moreover, the DCD network incorporates a novel contrastive pretext task based on local sub-segments, enhancing its ability to discriminate between anomalous and adjacent normal intervals. This improvement significantly increases the network's sensitivity to anomaly boundaries and its responsiveness across the entire anomaly range. Consequently, our approach can not only detects the occurrence of manipulation more timely but also localize their duration with greater accuracy.

Fig. 4 shows the overall architecture of the DCD network. It consists of two branches of annotation-free pretext tasks. They share the same representation encoder (the Transformer Body in Fig. 2), utilizing their respective pseudo labels and self-supervised learning schemes to train the network simultaneously. Among them, Pretext Task 1 uses representation learning of the entire time series to mine the connection between abnormal and normal points from a global perspective. Meanwhile, Pretext Task 2 accomplishes self-supervised training based on contrastive learning, focusing more on the local representation learning of abnormal areas. The purpose of Pretext Task 2 is to further explore the differences and uniqueness of the characteristics between anomaly segments and their adjacent normal segments.

1) Pretext Task 1: Global Representation Learning: To implement self-supervised learning, we need to design pseudo labels for the original unlabeled data and use them to provide supervisory signals for the pretext task during training. For Pretext Task 1, first, we utilize the synthetic anomalies

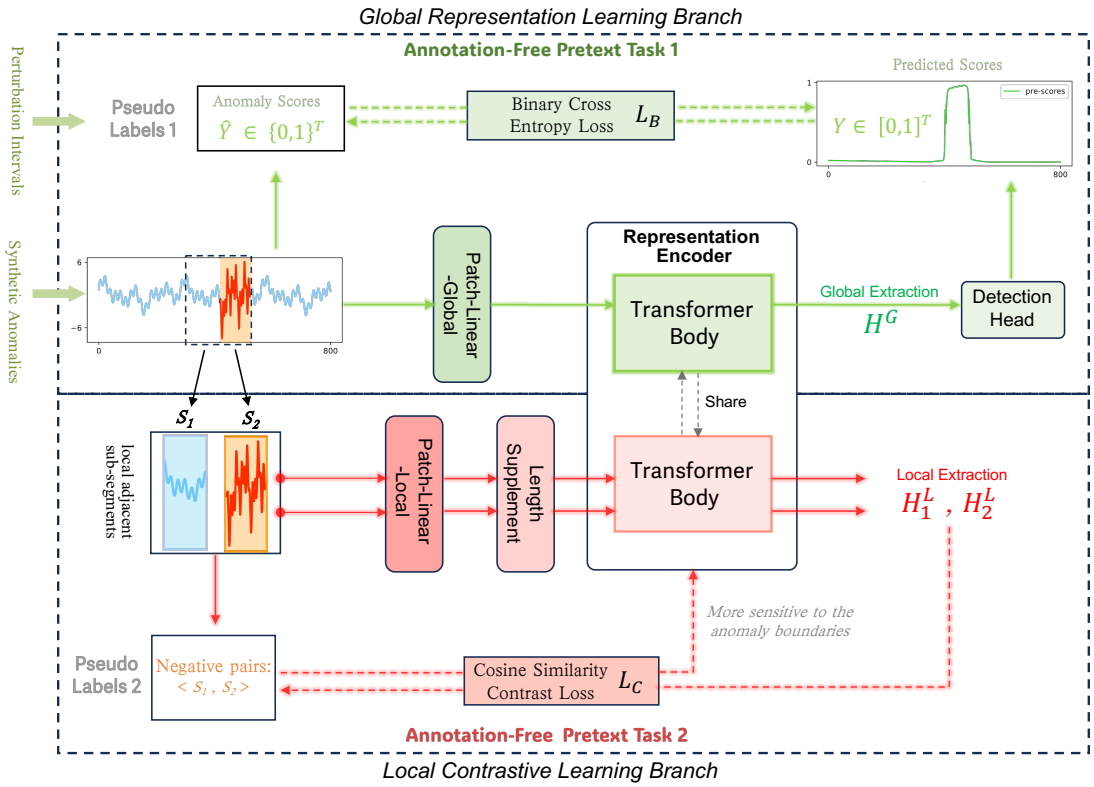

Fig. 4. The overall architecture of our Dual-branch Contrastive Detection (DCD) network.

Self-$Y \in \mathbb{R}^{N_s \times T}$ and their corresponding perturbation intervals to create the binary Pseudo Label 1 $\widehat{Y} \in \{0,1\}^{N_s \times T}$. N_s is the number of synthetic anomalies, while T is their length. $\widehat{Y}$ represents the anomaly score of each time point in the sample. The value is equal to 1 at the positions corresponding to the anomaly intervals, and 0 elsewhere. Second, we use a Patch-Linear Embedding module and a Transformer Body to model the entire time series, obtaining the Global Extraction Results H^G. Third, the Detection Head outputs the predicted anomaly scores $Y \in [0,1]^{N_s \times T}$ based on H^G. Lastly, we complete the training in Pretext Task 1 by optimizing the below Binary Cross Entropy Loss L_B.

$$L_B = -\frac{1}{N_s}\sum_{i=1}^{N_s}\left[\widehat{Y}_i log(Y_i) + (1-\widehat{Y}_i) log(1-Y_i)\right]. \quad (15)$$

Compared to conventional linear embedding before Transformer, a patch-wise linear projector [34] is more helpful for modeling the continuous temporal context of time series data. In detail, the Patch-Linear module we employed first splices P neighbouring points along the original channel dimension d to create a patch. Thus, the input time series $X^{'} \in \mathbb{R}^{T \times d}$ are patched as $X^{'} \in \mathbb{R}^{N \times (P \times d)} = \mathbb{R}^{N \times d_p}$, where N is the number of patches. Then, a linear projection operation is applied in the patched channel dimension d_p, and the shape of output is $z^G \in \mathbb{R}^{N \times d_E}$.

Finally, the dependencies among patches are modeled by Transformer Body to obtain H^G. This representation encoder is stacked by L identical layers, each of which mainly consists of a multi-headed self-attention (MHA) module followed by a multi-layer perceptron (MLP) block. We also adopt pre-norm residual connections in our Transformer Body, which can produce more stable gradients [31]. In summary, the representation results are computed as:

$$\tilde{z}_l = MHA(LayerNorm(z_{l-1})) + z_{l-1}, 1 \le l \le L \quad (16)$$

$$z_l = MHA(LayerNorm(\tilde{z}_l)) + \tilde{z}_l, 1 \le l \le L \quad (17)$$

where $z_0 = z^G$ and $H^G = z_L \in \mathbb{R}^{N \times d_E}$ in Pretext Task 1.

2) Pretext Task 2: Local Contrastive Learning: In real-world time series anomaly detection tasks, if a model can be adept at distinguishing between anomalous intervals and their adjacent normal intervals, particularly being sensitive to the boundaries between these two types of intervals, it would significantly enhance the performance of anomaly detection. This capability signifies a substantial improvement in detecting the occurrence of manipulations in a timely manner and in accurately localizing their duration.

To achieve the above objectives, we construct the Pretext Task 2 based on contrastive learning, focusing on the local areas of anomaly. Contrastive learning is a popular self-supervised learning strategy, where samples with different attributes are organized into negative pairs. The process of labelling samples as $\langle negative \rangle$ pairs provides pseudo labels for self-supervised learning. Following this, the training goal of the contrastive pretext task is to urge the feature encoder

to push the embeddings for negative pairs apart from each other. This will help downstream tasks to better identify and differentiate between these two types of samples.

In our task, as shown in the lower half of Fig. 4, we individually segment the anomalous fragments and name them S_a. Simultaneously, we also segment the equally long normal fragments adjacent to S_n and name them S_1. Based on the discussion above, we use these two new samples to construct negative pairs $\langle Sn, Sa \rangle$. Subsequently, synchronising with Pretext Task 1 and sharing the same representation encoder (i.e., Transformer Body), Pretext Task 2 trains our DCD based on contrastive representation learning.

In detail, first, these two time series segments are also projected into embedded features $z_n^L, z_a^L \in \mathbb{R}^{N^L \times d_E}$ with a Patch-Linear Embedding module. Second, in order to align with the required shape of representation encoder, a Length Supplement module extends the shape of these features to $\mathbb{R}^{N \times d_E}$ by the interpolation operation. Third, the shared Transformer Body utilizes z_n^L and z_a^L to calculate the Local Extraction Results $H_n^L, H_a^L \in \mathbb{R}^{N \times d_E}$ of these two sub-segments. Finally, we use $S_{neg}(H_n^L, H_a^L)$ to denote the cosine similarity of the extraction results of $\langle negative \rangle$ pairs, and input them into the contrastive loss function L_C to complete the training in Pretext Task 2:

$$S_{neg}(H_n^L, H_a^L) = \frac{\sum_{t=1}^{N} H_{n,t} \times H_{a,t}}{\sqrt{\sum_{t=1}^{N} (H_{n,t})^2} \times \sqrt{\sum_{t=1}^{N} (H_{a,t})^2}} \quad (18)$$

$$L_C = -\log(1 - \text{Sigmoid}(S_{neg}(H_n^L, H_a^L)/\tau) + \epsilon) \quad (19)$$

where τ is the temperature parameter controlling the sharpness of the similarity distribution in contrastive learning [42], ϵ is a small constant for numerical stability.

By continuously optimising the total loss L_{total} below, the training of Pretext Task 1 and Pretext Task 2 are performed simultaneously. Their main goal is to jointly learn a powerful representation encoder for time series anomaly detection:

$$L_{total} = L_B + L_C \quad (20)$$

3) Inference: The DCD's inference process will only apply Pretext Task 1, which can use the Detection Head to predict the anomaly score for each time point in the test set. An anomaly threshold will be set, and time points with anomaly scores exceeding this threshold are identified as anomalies. The specific value of the anomaly threshold is determined by a validation set.

Although the inference process does not use Pretext Task 2, during the self-supervised dual-branch training above, Pretext Task 2 has further improved the ability of our representation encoder to clearly distinguish anomalous points from other normal points in the input time series. The distinguishability of the representation results of the anomalous intervals and their adjacent normal intervals has also been further improved, which will greatly benefit the performance of the downstream anomaly detection head.

IV. Experiment

A. Datasets and Experimental Setting

Our evaluation utilises a unique dataset of Chinese stock market manipulation cases derived from regulatory enforcement actions by the China Securities Regulatory Commission (CSRC). This dataset addresses a critical gap in manipulation detection research, where labelled datasets are extremely scarce due to the clandestine nature of manipulative activities [11].

Dataset Construction: We systematically collected 164 regulatory reports documenting illegal trading activities in Chinese markets from 2017-2020. Following rigorous screening for annotation precision and temporal accuracy, we retained 25 cases across 10 stocks with reliable manipulation time windows. These cases encompass multiple manipulation types documented in the academic literature, including wash trading, pump-and-dump schemes, matched trading, cornering strategies, and spoofing [3].

Chinese Market Context: The Chinese equity market provides an ideal laboratory for manipulation detection research due to several unique characteristics. First, the market exhibits higher manipulation prevalence compared to mature markets, partly due to the large retail investor base and regulatory development stage [24]. Second, CSRC enforcement actions provide unusually detailed documentation of manipulation methods and timeframes, enabling precise labelling of anomalous periods.

Third, Chinese markets feature distinctive microstructure elements including daily price limits (typically ±10%), T+1 settlement, and concentrated trading sessions, which influence manipulation strategies and detection requirements. The predominance of retail investors (approximately 80% of trading volume) creates different information dynamics compared to institutionally-dominated markets, potentially affecting manipulation effectiveness and detection patterns.

Manipulation Types: Our dataset includes diverse manipulation strategies: (1) *Wash trading* involving self-dealing to create artificial volume; (2) *Pump-and-dump* schemes combining price inflation with coordinated selling; (3) *Matched trading* using pre-arranged transactions; (4) *Spoofing* through deceptive order placement and cancellation; and (5) *Cornering* via market concentration strategies. This diversity enables robust evaluation across different manipulation mechanisms.

B. Datasets and Experimental Setting

We evaluate our approach through a real-world case study utilizing a unique dataset of Chinese stock market manipulation cases. This dataset, derived from 164 regulatory reports issued by the Chinese Securities Commission, addresses empirical challenges posed by inexact and incomplete annotations. These reports document illegal activities in the Chinese stock market spanning from 2017 to 2020. After meticulous screening, we collect 25 cases with precise and reliable annotations of the manipulation time window across 10 stocks. They encompass multiple significant types of financial market manipulations, such as wash trading, pump and dump, matched trading, cornering the market, spoofing, and so on.

TABLE I
PERFORMANCE COMPARISON OF DIFFERENT ANOMALY DETECTION METHODS ON THE METRICS OF PRECISION. THE BEST RESULTS ARE IN **BOLD**. GIVEN THAT A SMALLER VALUE OF THE FALSE ALARM RATE (FAR) INDICATES BETTER DETECTION PERFORMANCE, WHICH IS CONTRARY TO OTHER METRICS, WE EXCLUDE FAR WHEN CALCULATING THE AVERAGE VALUES OF THE EVALUATION METRICS ABOVE. THE AVERAGE RESULTS ARE DENOTED AS AVG(W/O FAR).

Methods	**Metrics of Precision**				
	Precision	Recall	F1 Score	Accuracy	FAR
Isolation Forest	0.1655	0.1655	0.1655	0.7242	0.1652
KNN	0.2667	0.1986	0.2276	0.7773	0.1081
LSTM-Autoencoder	0.1915	0.1915	0.1915	0.7328	0.1600
TCN-Autoencoder	0.2648	0.2648	0.2648	0.7570	0.1455
Z-Score	0.2584	0.1277	0.1709	0.7953	0.0725
COPOD	0.1891	0.1891	0.1891	0.7320	0.1605
Matrix Profile	0.5845	0.1962	0.2938	0.8441	0.0276
Ensemble	0.1942	0.1915	0.1929	0.7352	0.1572
Sliding Window VAE	0.3857	0.3830	0.3843	0.7973	0.1207
Transformer-Autoencoder	0.2270	0.2270	0.2270	0.7445	0.1530
SDFM(ours)	**0.7759**	**0.4225**	**0.5471**	**0.9255**	**0.0145**
Methods	**Metrics of Precision**				
	IoU	MCC	ROC-AUC	PR-AUC	Avg(w/o FAR)
Isolation Forest	0.0902	0.0003	0.5001	0.2344	0.2557
KNN	0.1284	0.1023	0.5452	0.2988	0.3181
LSTM-Autoencoder	0.1059	0.0315	0.5157	0.2583	0.2773
TCN-Autoencoder	0.1526	0.1192	0.5596	0.3255	0.3385
Z-Score	0.0934	0.0748	0.5276	0.2651	0.2892
COPOD	0.1044	0.0286	0.5143	0.2561	0.2753
Matrix Profile	0.1722	0.2736	0.5843	0.4568	0.4257
Ensemble	0.1067	0.0345	0.5171	0.2597	0.2790
Sliding Window VAE	0.2379	0.2630	0.6311	0.4353	0.4397
Transformer-Autoencoder	0.1280	0.0739	0.5370	0.2908	0.3069
SDFM(ours)	**0.4388**	**0.5384**	**0.8802**	**0.6258**	**0.6443**

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT ANOMALY DETECTION METHODS ON THE METRICS OF TIMELINESS. THE BEST RESULTS ARE IN **BOLD**. A LOWER MEAN DETECTION DELAY (MDD) SIGNIFIES MORE TIMELY DETECTION, WHEREAS A HIGHER EARLY DETECTION SCORE (EDS) IS PREFERABLE.

Methods	**Metrics of Timeliness**	
	MDD	EDS
Isolation Forest	14.8750	0.9182
KNN	5.8125	0.9661
LSTM-Autoencoder	16.9375	0.9070
TCN-Autoencoder	11.1250	0.9412
Z-Score	6.0625	0.9642
COPOD	16.3125	0.9110
Matrix Profile	14.7500	0.9202
Ensemble	18.8750	0.8966
Sliding Window VAE	14.9375	0.9195
Transformer-Autoencoder	12.0000	0.9302
SDFM(ours)	**5.4400**	**0.9835**

Regarding the experimental settings, the following protocols are adopted: All experiments are conducted using PyTorch [43] on a NVIDIA A5000 24GB GPU. Optimization was carried out using the Adam optimizer [44] with an initial learning rate of 10^{-4}. We also employ the early-stop mechanism to prevent overfitting during training. The default number of synthetic anomalies generated is 1000, with 80% being used for training and the remaining 20% being used for validation. The test set comprises the aforementioned 25 real-world manipulation cases. In addition, we employ db2 [41] as the wavelet function of DWT in this study.

C. Baselines and Evaluations

We compare our SDFM with the following baseline methods proposed in recent years, which are popular in previous studies of manipulation detection and time series anomaly detection tasks:

- Isolation Forest [45]: detects anomalies by recursively partitioning data with randomly selected features and split points, where anomalies require fewer partitions to isolate than normal points.
- K-Nearest Neighbors (KNN) [46]: assigns anomaly scores based on the distance to the k-th nearest neighbor. This non-parametric method effectively captures local density variations without making distributional assumptions about the data generating process.
- LSTM Autoencoder [47]: employs a sequence-to-sequence architecture with a bottleneck structure to learn compressed representations of normal temporal patterns, using reconstruction error as the anomaly indicator.
- Temporal Convolutional Network (TCN) Autoencoder [48]: utilizes dilated causal convolutions to efficiently model both local and global temporal patterns with fewer

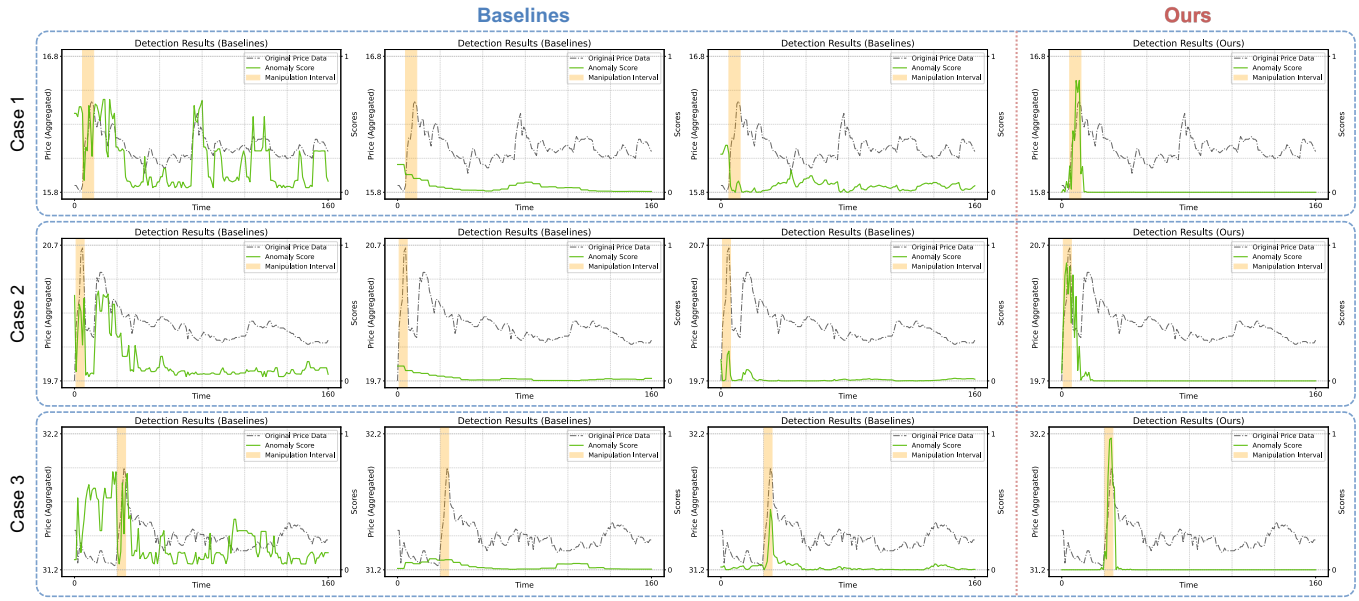

Fig. 5. The visualization of detection performance. The region masked in orange represents the manipulation interval, the gray curve represents the change of original stock price, the green curve is the anomaly scores predicted by different methods.

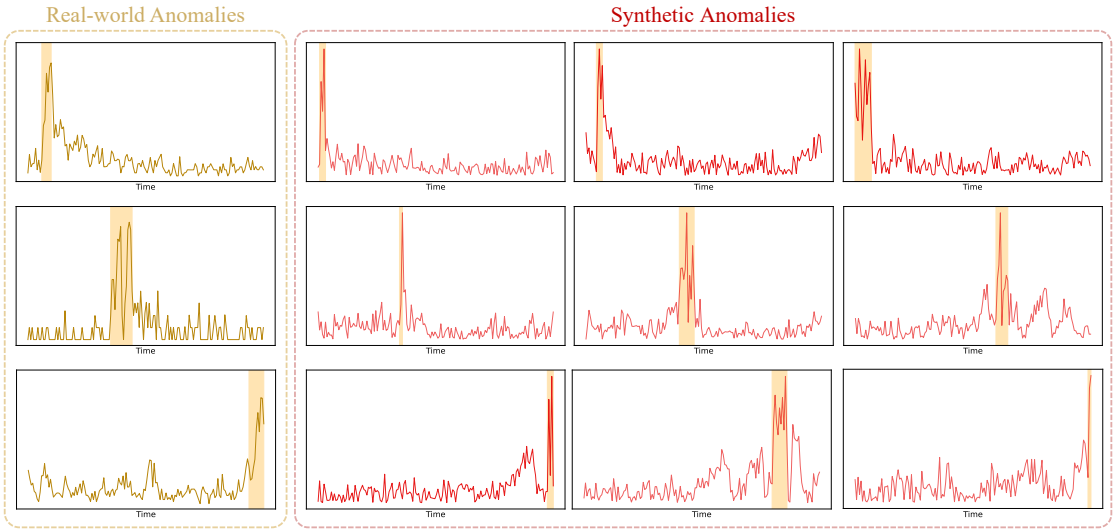

Fig. 6. The visualization of domain-specific representations (left) extracted from real-world manipulation cases and the synthetic anomalies (right). This can intuitively show that they have a high degree of similarity.

parameters than recurrent architectures. Anomalies are identified through reconstruction errors.

- Z-Score Detection [49]: applies statistical process control by measuring standardized deviations from local mean and standard deviation within a sliding window.
- Copula-Based Outlier Detection (COPOD) [50]: leverages empirical copula functions to model joint distributions without parametric assumptions, quantifying the extremeness of data points through tail probabilities.
- Matrix Profile [51]: constructs a meta time series recording the distance between each subsequence and its nearest non-overlapping match, identifying subsequences with unusually high minimum distances as anomalies.
- Ensemble Detection [52]: combines multiple heterogeneous anomaly detectors through techniques such as weighted averaging or majority voting to enhance stability and accuracy.
- Sliding Window Variational Autoencoder (VAE) [53]: applies variational inference to time series subsequences, encoding windows into latent distributions to quantify both reconstruction error and uncertainty.
- Transformer Autoencoder [54]: employs self-attention mechanisms to capture long-range dependencies between distant time points, enabling efficient parallel processing of entire sequences. It also identifies anomalies based on reconstruction errors in our task.

In terms of evaluation criteria, we utilize 9 precision metrics and 2 timeliness metrics to assess the performance of the

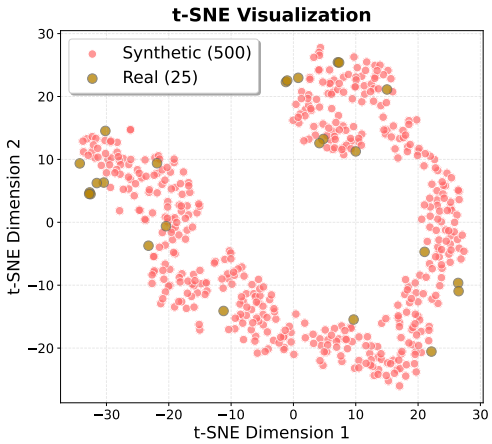

Fig. 7. The t-SNE visualization of synthetic anomalies (red dots) and domain-specific representations for our real-world manipulation cases (golden dots).

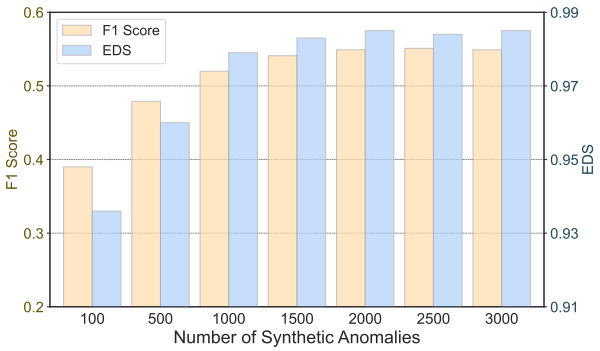

Fig. 8. The manipulation detection performance with different number of synthetic anomalies.

different methods. They provide a comprehensive and in-depth evaluation of the manipulation detection results.

The 9 precision metrics include Precision, Recall, F1 Score, Accuracy, False Alarm Rate (FAR), Intersection over Union (IoU), Matthews Correlation Coefficient (MCC), ROC-AUC, and PR-AUC.

The 2 timeliness metrics include the Mean Detection Delay (MDD) and the Early Detection Score (EDS). The former quantifies the average time lag between the onset of an anomaly and its detection by the model. The smaller its value is, the better the timeliness of the detection is. The later evaluates the timely identification of anomalies by assigning higher values to early detections through an exponential decay function. The calculation of these two metrics can be expressed as:

$$\text{MDD} = \frac{1}{N_a}\sum_{i=1}^{N_a}\max(0, t_{detect,i} - t_{start,i}) \tag{21}$$

where N_a is the number of anomalies, $t_{detect,i}$ is the time of detection for anomaly i, and $t_{start,i}$ is the actual start time of anomaly i.

$$\text{EDS} = \frac{1}{N_a}\sum_{i=1}^{N_a} e^{-\alpha \cdot \frac{delay_i}{L}} \tag{22}$$

where α is a decay parameter that controls the severity of the delay penalty, $delay_i$ is the detection delay for anomaly i, and L is the sequence length.

In addition, an evaluation technique known as Point Adjustment (PA) [55] has become popular in time series anomaly detection tasks in recent years. Specifically, if any observation in the ground truth abnormal segment is correctly detected, all observations in the segment are considered to be correctly detected. Given that it may lead to artificially inflated metrics, we will not use this calculation method in all experiments.

D. Main results

Table I offers a comprehensive comparison of our approach against ten baseline methods across various precision metrics. SDFM consistently surpasses all baselines in every evaluation metric, achieving the highest Precision, Recall, and F1 Score by substantial margins relative to the next best-performing methods.

Financial Surveillance Implications: The 77.59% precision rate represents a significant advancement for practical market surveillance, where false positives impose substantial costs on compliance teams and regulatory resources. Traditional rule-based systems often generate false alarm rates exceeding 95% [11], making our 1.45% false alarm rate a material improvement for real-world deployment. This precision level enables regulatory authorities to focus investigative resources on genuine manipulation cases rather than being overwhelmed by spurious alerts.

Economic Impact: The superior IoU (43.88%) and MCC (53.84%) scores demonstrate SDFM's ability to precisely localise manipulation intervals, crucial for determining the scope of market harm and calculating appropriate penalties. In Chinese regulatory cases, precise temporal boundaries are essential for identifying affected investors and calculating restitution, where imprecise detection can lead to either under-compensation of victims or excessive penalties for violators [24].

The ROC-AUC (88.02%) and PR-AUC (62.58%) scores indicate robust performance across different detection thresholds, enabling flexible deployment based on regulatory priorities. During high-volatility periods, regulators may prefer higher sensitivity settings, whilst routine surveillance may emphasise precision to minimise investigative burden.

Table II demonstrates SDFM's superior timeliness performance, achieving the lowest Mean Detection Delay (5.44) and highest Early Detection Score (0.9835) among all tested methods.

Market Impact Mitigation: The 5.44 mean detection delay represents approximately 5.5 time periods in our aggregated data, translating to rapid identification of manipulation schemes before significant market distortion occurs. Research on pump-and-dump schemes indicates that early detection within the first day can prevent price distortions exceeding 20% and protect substantial investor capital [2]. Our framework's early detection capability could prevent millions in investor losses by enabling swift regulatory intervention.

TABLE III
THE ABLATION STUDY OF DOMAIN-SPECIFIC FEATURES.

RO_B	RO_S	OCR_B	OCR_S	Metrics of Precision							Metrics of Timeliness	
				F1 Score	Accuracy	FAR	IoU	MCC	ROC-AUC	PR-AUC	MDD	EDS
✓	✗	✗	✗	0.5010	0.8972	0.0196	0.4002	0.4962	0.8266	0.5740	5.7221	0.9702
✗	✓	✗	✗	0.5036	0.9007	0.0194	0.4003	0.4960	0.8251	0.5761	5.7202	0.9716
✓	✓	✗	✗	0.5102	0.9096	0.0181	0.4106	0.5122	0.8372	0.5885	5.6002	0.9762
✓	✓	✓	✗	0.5311	0.9162	0.0165	0.4256	0.5275	0.8565	0.6082	5.4996	0.9799
✓	✓	✗	✓	0.5306	0.9169	0.0168	0.4251	0.5271	0.8551	0.6091	5.4910	0.9811
✓	✗	✓	✗	0.5186	0.9109	0.0175	0.4196	0.5201	0.8412	0.5962	5.5316	0.9781
✗	✓	✗	✓	0.5187	0.9102	0.0178	0.4199	0.5205	0.8406	0.5971	5.5312	0.9786
✓	✓	✓	✓	**0.5471**	**0.9255**	**0.0145**	**0.4388**	**0.5384**	**0.8802**	**0.6258**	**5.4400**	**0.9835**

TABLE IV
THE ABLATION STUDY OF OUR DCD NETWORK.

GP	LP	CLB	Metrics of Precision							Metrics of Timeliness	
			F1 Score	Accuracy	FAR	IoU	MCC	ROC-AUC	PR-AUC	MDD	EDS
✗	✗	✗	0.5126	0.9093	0.0179	0.4112	0.5125	0.8393	0.5910	5.6822	0.9746
✓	✗	✗	0.5185	0.9112	0.0173	0.4190	0.5186	0.8402	0.5966	5.6105	0.9769
✗	✗	✓	0.5356	0.9174	0.0158	0.4303	0.5296	0.8682	0.6151	5.5008	0.9812
✓	✗	✓	0.5422	0.9209	0.0149	0.4355	0.5326	0.8761	0.6202	5.4696	0.9826
✗	✓	✓	0.5416	0.9201	0.0150	0.4351	0.5322	0.8750	0.6210	5.4722	0.9821
✓	✓	✓	**0.5471**	**0.9255**	**0.0145**	**0.4388**	**0.5384**	**0.8802**	**0.6258**	**5.4400**	**0.9835**

Regulatory Response Time: The Early Detection Score of 0.9835 indicates that SDFM identifies 98.35% of manipulations near their onset, providing regulatory authorities with timely alerts for investigation and enforcement action. This rapid detection aligns with modern market surveillance requirements where algorithmic trading can amplify manipulation effects within minutes, making early intervention critical for market stability.

Comparative Context: Traditional econometric approaches (KNN, Z-Score) achieve detection delays of 5-6 periods but with significantly lower precision, while more sophisticated methods (Matrix Profile, VAE) sacrifice timeliness for accuracy. SDFM uniquely combines rapid detection with high precision, addressing the fundamental trade-off in financial surveillance between speed and accuracy that has historically plagued market oversight systems.

E. Framework Analysis

1) Visualization of Detection Performance: Fig. 5 provides crucial insights into SDFM's detection behaviour across diverse manipulation scenarios. The visualizations reveal four distinct manipulation cases representing different schemes prevalent in Chinese markets: coordinated pump-and-dump operations, spoofing activities, and wash trading patterns.

Manipulation Pattern Recognition: SDFM's anomaly scores (green curves) demonstrate superior responsiveness to manipulation onset, generating elevated signals immediately upon scheme initiation. This early signal generation is particularly evident in Cases 1 and 3, where SDFM detects coordination patterns that baseline methods miss entirely. The sharp signal increases align with theoretical expectations of rush order clustering and abnormal cancellation patterns characteristic of coordinated manipulation [36].

Signal Persistence and Boundary Detection: The sustained high anomaly scores throughout manipulation intervals demonstrate SDFM's ability to track ongoing schemes rather than generating sporadic alerts. This persistence is crucial for regulatory investigation, as it provides clear evidence of manipulation duration for penalty calculation and victim identification. The sharp signal decay at manipulation termination (visible in Cases 2 and 4) indicates precise boundary detection, essential for determining the scope of market harm.

Financial Market Context: The price movements (gray curves) show typical manipulation patterns documented in academic literature: gradual price inflation during pump phases followed by sharp declines. SDFM's signals align closely with these price dynamics, suggesting the framework captures the underlying manipulative trading behaviour rather than merely responding to price volatility. This alignment validates our domain-specific feature engineering approach, demonstrating that Rush Order and Order Cancellation Ratio features effectively amplify manipulation signals whilst filtering normal market noise.

Practical Deployment Implications: The consistent signal quality across different manipulation types suggests SDFM's robustness for real-world deployment. Unlike baseline methods that show erratic responses or miss manipulations entirely, SDFM provides reliable, interpretable signals that compliance teams can act upon with confidence. This reliability is essential for regulatory acceptance, as inconsistent detection performance undermines investigator confidence and delays enforcement actions.

2) Realism of Synthetic Anomaly: We evaluate the fidelity of our synthetic anomaly generation by comparing domain-specific representations from three real-world manipulation cases with nine randomly selected synthetic anomalies. The

real cases represent manipulations occurring during early, middle, and late trading phases, capturing the temporal diversity of manipulation strategies documented in the literature [2].

Manipulation Type Diversity: Fig. 6 visualizes the distinctive anomaly patterns across different manipulation schemes. The domain-specific representations demonstrate clear differentiation between manipulation types prevalent in Chinese markets: Pump-and-Dump schemes showing characteristic volume clustering, Spoofing activities exhibiting rapid order cancellation patterns, Wash Trading displaying artificial transaction coordination, and Fake Liquidity manipulation revealing deceptive order book dynamics [37]. This diversity reflects the sophisticated nature of modern market manipulation, where perpetrators employ multiple coordinated strategies to evade detection.

Synthetic Fidelity Assessment: Our synthetic anomalies successfully replicate the statistical properties and temporal patterns of real manipulation cases. The frequency-domain synthesis approach preserves the multi-scale characteristics essential to manipulation detection, ensuring that synthetic training samples contain the same Rush Order clustering and Order Cancellation ratio patterns observed in actual schemes [36]. This fidelity is crucial for self-supervised learning effectiveness, as poor-quality synthetic data would lead to detection models that fail to generalise to real-world manipulation patterns.

Training Data Augmentation: The ability to generate unlimited realistic synthetic anomalies addresses a fundamental challenge in financial surveillance: the scarcity of labeled manipulation data. Traditional supervised approaches are severely limited by the small number of confirmed manipulation cases available for training [11]. Our synthetic generation capability enables robust model training whilst preserving the unique characteristics of different manipulation strategies, providing SDFM with comprehensive exposure to manipulation patterns that would be impossible to achieve using historical cases alone.

Validation Against Market Microstructure Theory: The synthetic anomalies align with established market microstructure principles governing informed trading behavior [15]. The preserved temporal clustering patterns reflect the strategic timing considerations of manipulators, whilst the maintained volume and cancellation characteristics capture the operational constraints faced by coordinated trading groups. This theoretical consistency validates our frequency-based synthesis approach and ensures that SDFM learns manipulation patterns grounded in sound financial theory rather than spurious statistical correlations.

At the same time, we further utilize t-SNE algorism to quantitatively analyze the realism of synthetic anomaly. Fig. 7 visualizes the low-dimensional embedding results of t-SNE between our synthetic anomalies (red dots) and the domain-specific representations for our real-world manipulation cases (golden dots). They manifest not only a high degree of global distribution consistency but also remarkably close local proximity, strongly indicating substantial similarity between these two types of anomalous samples and corroborating the effectiveness of our synthesis approach.

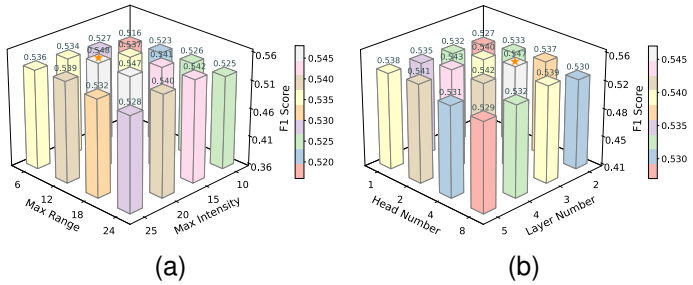

Fig. 9. Parameter sensitivity analysis. The orange star ⋆ marks the configuration achieving the highest F1 Score.

Additionally, Fig. 8 analyzes the impact of the number of input synthetic anomalies during network training. As their number increases, the manipulation detection performance of SDFM is evidently enhanced, which also validates the efficacy of these synthetic training samples. We use the F1 Score and EDS as representatives of the precision and timeliness metrics, respectively. The detection performance peaks at around 2,000 samples. Notably, collecting thousands of labeled anomaly samples to automatically train a detector is often extremely difficult for conventional detection methods, while our approach can readily provide the model with an unlimited supply of required training data.

3) Ablation Study: In order to evidently demonstrate the contributions of our Amplification Component, we conduct a comprehensive ablation study on all domain-specific features within it. Since OCR_B and OCR_S serve as the scaling factors our fusion process, the ablation study will not apply these two types of features solely. Meanwhile, considering that F1 Score can represent the harmonic mean of Precision and Recall, we omit the comparison of Precision and Recall here for simplicity. The experiment results are illustrated in Table III, where the first 5 rows represent the process of incrementally integrating different features for the domain-specific representations. As the number of domain-specific features increase, there is an obvious trend of improvement within the metrics of both precision and timeliness, demonstrating the effectiveness of each feature and their fusion process. It is noteworthy that even when using only one domain-specific feature, our method still outperforms the best performing baseline method significantly. In addition, the experiments in 6-th and 7-th row are mainly used for investigating the differences between the features of buy-side and sell-side. Their similar performance verify that both these two types of orders play crucial role in extracting and representing the manipulation patterns.

At the same time, we also conduct an ablation study for our DCD network to investigate the effect of each component within it. Table IV gradually examines the contributions of the Global Patch-Linear module (GP), the Local Patch-Linear module (LP), and the Local Contrastive Learning Branch (CLB). Among them, LP is applied only when CLB exists. The GP and LP modules will be replaced by conventional linear embedding when they are absent. The results in Table IV demonstrate that the patch-linear operation in both self-supervised learning branches have a positive impact on the

manipulation detection of our SDFM. When introducing CLB, the improvement of our performance is most significant, which evidently illustrates the effectiveness of our contrastive learning scheme. In particular, CLB enables tremendous progress on timeliness metrics, strongly demonstrating its contribution in enhancing the network's sensitivity to the occurrence of manipulations.

4) Parameters Sensitivity: We also investigate the parameter sensitivity of our SDFM. Fig. 9(a) displays the impact of different max range p^{max} and max intensity A^{max} of perturbation during synthetic frequency distribution generation (as shown in Algorithm 3). The results indicate that the manipulation detection performance is optimal when these two critical perturbation parameters are set to moderate values. Both too small and too large perturbations compromise the resemblance between synthetic and real-world anomalies, consequently hindering the network's learning of authentic manipulation patterns. Meanwhile, Fig. 9(b) investigates the impact of two important Transformer architecture hyper-parameters within the DCD network: the number of attention heads and the number of stacked layers, as the performance of many deep neural networks is affected by them.

V. Conclusion

An increasing number of studies have focused on detecting market manipulation using financial time series of transaction data, see for example [1]–[3]. However, the outcomes often fall short of expectations and remain somewhat removed from practical deployment, due to the challenges of signal concealment, data sparsity and boundary vagueness. In this paper, we introduce a **S**elf-supervised **D**etection **F**ramework tailored to financial markets **M**anipulation, named **SDFM**.

We evaluate our approach using a real-world case study, comprising a unique dataset of Chinese stock market manipulation cases. Our framework achieves performance consistently superior to state-of-the-art methods across 11 metrics. Extensive quantitative and qualitative experiments demonstrate that our framework can generate a much faster and stronger response to real-world manipulation, offering new insights into financial market manipulation and establishing a powerful approach for detecting these anomalies.

References

[1] J.-L. Vila, "Simple games of market manipulation," *Economics Letters*, vol. 29, no. 1, pp. 21–26, 1989.

[2] R. K. Aggarwal and G. Wu, "Stock market manipulations," *The Journal of Business*, vol. 79, no. 4, pp. 1915–1953, 2006.

[3] T. J. Putniņş, "Market manipulation: a survey," *Journal of Economic Surveys*, vol. 26, no. 5, pp. 952–967, 2012.

[4] M. Kacperczyk and E. S. Pagnotta, "Legal risk and insider trading," *The Journal of Finance*, vol. 79, no. 1, pp. 305–355, 2024.

[5] M. Khomyn and T. J. Putniņš, "Algos gone wild: What drives the extreme order cancellation rates in modern markets?," *Journal of Banking & Finance*, vol. 129, p. 106170, 2021.

[6] L. Close, R. Kashef, *et al.*, "Combining artificial immune system and clustering analysis: A stock market anomaly detection model," *Journal of Intelligent Learning Systems and Applications*, vol. 12, no. 04, p. 83, 2020.

[7] T. Li, G. Kou, Y. Peng, and P. S. Yu, "An integrated cluster detection, optimization, and interpretation approach for financial data," *IEEE transactions on cybernetics*, vol. 52, no. 12, pp. 13848–13861, 2021.

[8] S. Khodabandehlou and A. H. Golpayegani, "Fifraud: unsupervised financial fraud detection in dynamic graph streams," *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 5, pp. 1–29, 2024.

[9] R. Bénabou and G. Laroque, "Using privileged information to manipulate markets: Insiders, gurus, and credibility," *The Quarterly Journal of Economics*, vol. 107, no. 3, pp. 921–958, 1992.

[10] J. Van Bommel, "Rumors," *The Journal of Finance*, vol. 58, no. 4, pp. 1499–1520, 2003.

[11] R. James, H. Leung, and A. Prokhorov, "A machine learning attack on illegal trading," *Journal of Banking & Finance*, vol. 148, p. 106735, 2023.

[12] C. Liu, S. He, H. Liu, and S. Li, "Treemil: A multi-instance learning framework for time series anomaly detection with inexact supervision," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7510–7514, IEEE, 2024.

[13] C. Zhang, G. Li, Y. Qi, S. Wang, L. Qing, Q. Huang, and M.-H. Yang, "Exploiting completeness and uncertainty of pseudo labels for weakly supervised video anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16271–16280, 2023.

[14] H. Hoeltgebaum, N. Adams, and C. Fernandes, "Estimation, forecasting, and anomaly detection for nonstationary streams using adaptive estimation," *IEEE Transactions on Cybernetics*, vol. 52, no. 8, pp. 7956–7967, 2021.

[15] L. R. Glosten and P. R. Milgrom, "Bid, ask and transaction prices in a specialist market with heterogeneously informed traders," *Journal of Financial Economics*, vol. 14, no. 1, pp. 71–100, 1985.

[16] H. He and J. Wang, "Differential information and dynamic behavior of stock trading volume," *The Review of Financial Studies*, vol. 8, no. 4, pp. 919–972, 1995.

[17] G. Liu, F. Xiao, C.-T. Lin, and Z. Cao, "A fuzzy interval time-series energy and financial forecasting model using network-based multiple time-frequency spaces and the induced-ordered weighted averaging aggregation operation," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 11, pp. 2677–2690, 2020.

[18] H. Rezaei, H. Faaljou, and G. Mansourfar, "Stock price prediction using deep learning and frequency decomposition," *Expert Systems with Applications*, vol. 169, p. 114332, 2021.

[19] M. Wang, Q. Wang, D. Hong, S. K. Roy, and J. Chanussot, "Learning tensor low-rank representation for hyperspectral anomaly detection," *IEEE Transactions on Cybernetics*, vol. 53, no. 1, pp. 679–691, 2022.

[20] P. Kumar and D. J. Seppi, "Futures manipulation with "cash settlement"," *The Journal of Finance*, vol. 47, no. 4, pp. 1485–1502, 1992.

[21] P. Hillion and M. Suominen, "The manipulation of closing prices," *Journal of Financial Markets*, vol. 7, no. 4, pp. 351–375, 2004.

[22] D. Diaz, B. Theodoulidis, and P. Sampaio, "Analysis of stock market manipulations using knowledge discovery techniques applied to intraday trade prices," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12757–12771, 2011.

[23] K. Golmohammadi, O. R. Zaiane, and D. Díaz, "Detecting stock market manipulation using supervised learning algorithms," pp. 435–441, 2014.

[24] X.-Q. Sun, H.-W. Shen, X.-Q. Cheng, and Y. Wang, "Detecting anomalous traders using multi-slice network analysis," *Physica A: Statistical Mechanics and its Applications*, vol. 473, pp. 1–9, 2017.

[25] M. Kacperczyk and E. S. Pagnotta, "Legal risk and insider trading," *The Journal of Finance*, vol. 79, no. 1, pp. 309–362, 2024.

[26] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," *arXiv preprint arXiv:2110.02642*, 2021.

[27] Z. Z. Darban, G. I. Webb, S. Pan, C. C. Aggarwal, and M. Salehi, "Deep learning for time series anomaly detection: A survey," *arXiv preprint arXiv:2211.05244*, 2022.

[28] Y. Jeong, E. Yang, J. H. Ryu, I. Park, and M. Kang, "Anomalybert: Self-supervised transformer for time series anomaly detection using data degradation scheme," *arXiv preprint arXiv:2305.04468*, 2023.

[29] C. Xiao, S. Chen, F. Zhou, and J. Wu, "Self-supervised few-shot time-series segmentation for activity recognition," *IEEE Transactions on Mobile Computing*, 2022.

[30] G. Pang, C. Shen, and A. Van Den Hengel, "Deep anomaly detection with deviation networks," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 353–362, 2019.

[31] E. Eldele, M. Ragab, Z. Chen, M. Wu, C.-K. Kwoh, X. Li, and C. Guan, "Self-supervised contrastive representation learning for semi-supervised time-series classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[32] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

[33] X. Zhang, Z. Zhao, T. Tsiligkaridis, and M. Zitnik, "Self-supervised contrastive pre-training for time series via time-frequency consistency," *Advances in Neural Information Processing Systems*, vol. 35, pp. 3988–4003, 2022.

[34] Y. Yang, C. Zhang, T. Zhou, Q. Wen, and L. Sun, "Dcdetector: Dual attention contrastive representation learning for time series anomaly detection," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3033–3045, 2023.

[35] A. S. Kyle, "Continuous auctions and insider trading," *Econometrica*, pp. 1315–1335, 1985.

[36] M. Khomyn and T. J. Putniņş, "Algos gone wild: What drives the extreme order cancellation rates in modern markets?," *Journal of Banking & Finance*, vol. 129, p. 106170, 2021.

[37] D. Cumming, S. Johan, and D. Li, "Exchange trading rules and stock market liquidity," *Journal of Financial Economics*, vol. 99, no. 3, pp. 651–671, 2011.

[38] K. Xiong, H. H. Iu, and S. Wang, "Kernel correntropy conjugate gradient algorithms based on half-quadratic optimization," *IEEE Transactions on Cybernetics*, vol. 51, no. 11, pp. 5497–5510, 2020.

[39] M. Davarifar, A. Rabhi, A. Hajjaji, E. Kamal, and Z. Daneshifar, "Partial shading fault diagnosis in pv system with discrete wavelet transform (dwt)," in *2014 International Conference on Renewable Energy Research and Application (ICRERA)*, pp. 810–814, IEEE, 2014.

[40] J. Yang and S.-T. Park, "An anti-aliasing algorithm for discrete wavelet transform," *Mechanical Systems and Signal Processing*, vol. 17, no. 5, pp. 945–954, 2003.

[41] H. Rajaguru and S. K. Prabhakar, "Time frequency analysis (db2 and db4) for epilepsy classification with lda classifier," in *2017 2nd international conference on communication and electronics systems (ICCES)*, pp. 708–711, IEEE, 2017.

[42] J. Mitrovic, B. McWilliams, and M. Rey, "Less can be more in contrastive learning," *NeurIPS Workshops*, pp. 70–75, 2020.

[43] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*, pp. 4393–4402, PMLR, 2018.

[44] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International conference on learning representations*, 2018.

[45] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*, pp. 413–422, IEEE, 2008.

[46] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 427–438, 2000.

[47] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," *arXiv preprint arXiv:1607.00148*, 2016.

[48] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[49] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[50] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "Copod: copula-based outlier detection," in *2020 IEEE international conference on data mining (ICDM)*, pp. 1118–1123, IEEE, 2020.

[51] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, and E. Keogh, "Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets," in *2016 IEEE 16th international conference on data mining (ICDM)*, pp. 1317–1322, Ieee, 2016.

[52] C. C. Aggarwal, "Outlier ensembles," in *Outlier Analysis*, pp. 185–218, Springer, 2016.

[53] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[54] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," *Advances in neural information processing systems*, vol. 32, 2019.

[55] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, *et al.*, "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *Proceedings of the 2018 world wide web conference*, pp. 187–196, 2018.