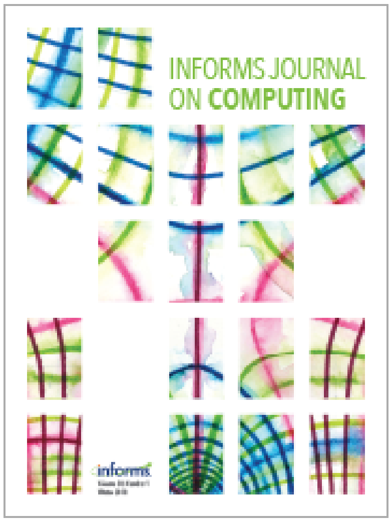


INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Deterring the Gray Market: Product Diversion Detection via Learning Disentangled Representations of Multivariate Time Series

Hao Lin, Guannan Liu, Junjie Wu, J. Leon Zhao

INFORMS JOURNAL ON COMPUTING
Articles in Advance, pp. 1–16
ISSN 1091-9856 (print), ISSN 1526-5528 (online)
https://pubsonline.informs.org/journal/ijoc

Deterring the Gray Market: Product Diversion Detection via Learning Disentangled Representations of Multivariate Time Series

Hao Lin,[a,b] Guannan Liu,[a,b,*] Junjie Wu,[a,b] J. Leon Zhao[c]

[a] School of Economics and Management, Beihang University, Beijing 100191, China; [b] Key Laboratory of Data Intelligence and Management, Ministry of Industry and Information Technology, Beijing 100191, China; [c] The Chinese University of Hong Kong, Shenzhen 518172, China
*Corresponding author

Contact: haolin@buaa.edu.cn, https://orcid.org/0000-0002-1921-3036 (HL); liugn@buaa.edu.cn, https://orcid.org/0000-0002-4532-7109 (GL); wujj@buaa.edu.cn, https://orcid.org/0000-0001-7650-3657 (JW); leonzhao@cuhk.edu.cn, https://orcid.org/0000-0002-0624-0254 (JLZ)

Received: May 21, 2022
Revised: March 1, 2023; August 13, 2023; October 8, 2023
Accepted: October 10, 2023
Published Online in Articles in Advance: December 6, 2023

https://doi.org/10.1287/ijoc.2022.0155

Abstract. A gray market emerges when some distributors divert products to unauthorized distributors/retailers to make sneaky profits from the manufacturers' differential channel incentives, such as quantity discounts. Traditionally, manufacturers rely heavily on internal audits to periodically investigate the flows of products and funds so as to deter the gray market; however, this is too costly given the large number of distributors and their huge volumes of orders. Owing to the advances in data analytics techniques, the ordering quantities of a distributor over time, which form multivariate time series, can help reveal suspicious product diversion behaviors and narrow the audit scope drastically. To that end, in this paper, we build on the recent advancement of representation learning for time series and adopt a sequence autoencoder to automatically characterize the overall demand patterns. To cope with the underlying entangled factors and interfering information in the multivariate time series of ordering quantities, we develop a disentangled learning scheme to construct more effective sequence representations. An interdistributor correlation regularization is also proposed to ensure more reliable representations. Finally, given the highly scarce anomaly labels for the detection task, an unsupervised deep generative model based on the learned representations of the distributors is developed to estimate the densities of distributions, which enables the anomaly scores generated through end-to-end learning. Extensive experiments on a real-world distribution channel data set and a larger simulated data set empirically validate our model's superior and robust performances compared with several state-of-the-art baselines. Additionally, our illustrative economic analysis demonstrates that the manufacturers can launch more targeted and cost-effective audits toward the suspected distributors recommended by our model so as to deter the gray market.

History: Accepted by Ram Ramesh, Area Editor for Data Science & Machine Learning.
Funding: This work was supported by the National Natural Science Foundation of China [Grants 72031001, 72301017, 72371011, and 72242101].
Supplemental Material: The software that supports the findings of this study is available within the paper and its Supplemental Information (https://pubsonline.informs.org/doi/suppl/10.1287/ijoc.2022.0155) as well as from the IJOC GitHub software repository (https://github.com/INFORMSJoC/2022.0155). The complete IJOC Software and Data Repository is available at https://informsjoc.github.io/.

Keywords: gray market • product diversion • multivariate time series • disentangled learning • anomaly detection

1. Introduction

The gray market (GM) refers to the practice in which authorized distributors resell the products ordered from manufacturers to unauthorized retailers, which creates a market outside the control of the original manufacturer (Antia et al. 2006). The gray market has become a prevailing phenomenon across many industries, including fashion, electronics, computers, etc., which naturally emerges when a manufacturer launches price discrimination among different distribution channels (Srivastava and Mateen 2020). For example, a manufacturer may launch quantity discounts toward the downstream distributors with large ordering quantities to incentivize sales. In this regard, distributors may take advantage of such quantity discounts by inflating their ordering quantities to benefit from the quantity discounts and then resell the products to unauthorized retailers with a higher price, which is often known as product diversion (Dasu et al. 2012). One of the world-renowned computer manufacturers, Hewlett-Packard Company (HP), is reported to encounter the gray market of its products. For example, a distributor of HP overstocked laptop computers under a quantity discount contract and diverted them to the gray market (Srivastava and Mateen 2020); HP has also filed lawsuits against its distributors because of the product diversion to the unauthorized channel (HP 2008). In addition, it is also found that several authorized distributors of China HP have formed alliances in the gray market, in which one distributor in the alliance may inflate its ordering quantities to qualify for the quantity discount, whereas the other distributors in the alliance conversely shrink their direct ordering quantities from the manufacturer. Then, the distributor with overstocked inventory diverts the excess inventory to the distributors in the alliance at the discounted price, and as a result, all of them can enjoy the larger discounts, which, however, hurts HP's profits and the benefits of its incentive program.

As a typical type of supply chain fraud (KPMG 2017), product diversion in the gray market has become a long-standing challenge for manufacturers and authorized distributors as well as end consumers (Myers 1999). It is reported that sales in the gray market of copyrighted works, including books, music, and movies, can be as high as $220 billion per year (Srivastava and Mateen 2020). Because of the disordered product diversion and inefficient price control among the distribution channels, the gray market may result in direct reduced profits and deterioration of channel relationships (Antia et al. 2006). Furthermore, the products diverted in the gray market may not have a warranty, which can hurt consumers' benefits. In view of this, manufacturers are seeking ways to deter the gray market. A periodic unannounced audit is deemed to be a common strategy to prevent the formation of a gray market (Antia et al. 2004, 2006). By monitoring the orders and sales of the distributors, the manufacturer can keep an eye on the product and fund flow, and hence, the market can be regulated in case of product diversion. However, it is still impractical to conduct internal audits for each distributor because they would usually incur huge effort and economic costs.

Modern information systems have equipped firms to deter the gray market in a more efficient way. Particularly, enterprise resource planning is a common information system in modern firms to record daily business activities, including orders, sales, and inventory (Askenaes and Westelius 2000). Thus, the records of ordering quantities for each downstream distributor are available to guide the firms to probe the evidence of product diversion via a data-driven method. As a matter of fact, the demand for each stock-keeping unit (SKU) may exhibit some temporal regularities, for example, seasonal fluctuations, market growth, and market recession. Thus, we can estimate the general demand patterns by tracking the time series of ordering quantities (TSOQ) for all the SKUs in the catalog with each SKU being a unique channel for the multivariate time series. If the gray market exists, distributors involved in product diversion would show contrastive patterns in their TSOQs compared with the normal distributors. For example, the distributor who inflates its ordering quantities may have persistent larger ordering quantities regardless of the seasonal effects, which deviates from the ordering quantities of normal distributors by and large.

Although extant research has been proposed to tackle anomaly detection in time series, the deterrence of the gray market from the perspective of time series has yet to receive adequate attention in academia and industries. In the lens of monitoring the TSOQ for product diversion detection, several critical challenges need to be addressed. First, prior anomaly detection for time series primarily focuses on extracting handcrafted features that can reflect the abnormality of the time series, which requires substantive domain knowledge. However, the ordering quantities of distributors with the aim of product diversion may not change abruptly, and such deviations from the normal distributors may last for a longer time period. For example, we may witness a gradual upward lift in the ordering quantities with persistent larger quantities than other distributors, which, however, cannot be captured from a single snapshot. Thus, handcrafted features extracted within a short time window may not be applicable in this sense. Second, each distributor may hold a unique assortment of SKUs with certain relations, such as substitution and complementarity, in between, which incur complex and entangled cross-channel relations in the time series. In addition, there exists a considerable amount of extreme values, such as sudden changes and zero quantities within the TSOQ, which may result from market fluctuations or the assortments planned by the distributors. Such disturbed information in TSOQ, however, can hinder the recognition of normal ordering patterns. Furthermore, the ordering quantities might also be correlated among the distributors. Concretely, the distributors may exhibit similar ordering patterns because of some systematic changes in the market, showing a positive correlation pattern, whereas a distributor with possible product diversion may exert distinctive correlations with others because of its inflation of ordering quantities.

To tackle the aforementioned challenges for detecting product diversion, we propose to learn more expressive representations of the TSOQs to reflect the overall characteristics of the distributors. Specifically, we adopt a sequence autoencoder (Sutskever et al. 2014) to represent the multivariate time series through a reconstruction procedure. Meanwhile, we design a disentangled learning layer with an adversarial training scheme to decouple the underlying intricate cross-channel relations between different SKUs, thereby reducing the negative impact of the interfering information in TSOQs. Consequently, TSOQs can be disentangled into several latent segments of more effective representations, each indicating distinct ordering patterns. In order to tailor the representations for the unsupervised setting of anomaly detection, we estimate the density of the learned representations via a Gaussian mixture model (GMM) (Zong et al. 2018) in which the representation learning and anomaly detection tasks are integrated in an end-to-end fashion with a unified objective function. Furthermore, to account for the correlation patterns between the distributors, we also enforce a constraint of correlations between the time series for learning more reliable representations.

We demonstrate the effectiveness of our proposed method on a real-world distribution channel data set collected from a famous computer manufacturer for which only a small proportion of distributors are identified as product diverters. Our empirical results show superior overall detection performance as measured by the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPRC) compared with the baseline methods, and the modeling modules, including the disentangled learning and the correlation regularization, are both shown to play vital roles in improving the model performances. We also conduct an economic analysis to demonstrate the cost-effectiveness of the proposed method. In addition, in order to further demonstrate the robustness of our proposed method, we generate distributors' ordering activities with agent-based simulations and inject anomalous product diversion activities with different proportions in both training and test sets in which our model gains consistently better detection performance under different settings.

2. Literature Review

The paper is related to three streams of literature, including deterrence of the gray market, anomaly detection from time series, and time series representation learning.

2.1. Deterrence from the Gray Market

The GM is a long-standing problem and has been studied from both analytical and empirical perspectives in the past several decades (Cespedes et al. 1988). Prior studies mainly focus on the causes and effects of GM. For example, Ahmadi et al. (2015) study the mechanism of the emergence of GM under demand uncertainty. Zhang and Feng (2017) propose a pricing model for the firm when faced with the gray market and further study how GM can influence consumer demands. Srivastava and Mateen (2020) analyze the performance of the supply chain contract in the presence of a gray market and the impact on price, quantity, etc. In addition, Ahmadi and Yang (2000) point out the challenges and harms of GM.

To deter the gray market, different countermeasures have been studied extensively (Zhao et al. 2021). Antia et al. (2006) examine whether and how enforcement deters the gray market and also point out that the manufacturers have to invest much in the systems to make the enforcement effective. Other research uses incentive mechanisms, such as pricing models, to deter GM. For instance, Zhang (2016) finds that the manufacturer can use a consumer rebate strategy, which gives payment directly to consumers to fight against the gray market. Su and Mukhopadhyay (2012) propose to design contracts to manage the distribution network and further counter GM.

Except for the mechanism design, recent research seeks technical ways to deter the gray market by tracking the flows of products with the new technology. For example, Ding et al. (2019) design an internet of things (IoT)-based traceability system to track the products sold to the distributors. de Boissieu et al. (2021) design a blockchain for tracing products. Though tracing products is deemed effective for monitoring suspicious product diversion, it might take huge costs to construct the information technology (IT) infrastructure. With the recent development of big data and machine learning techniques, several studies have begun to tackle the supply chain problems via forecasting or anomaly detection based on the data generated in the supply chain, including sales data (Thomassey 2010) and inventory data (Roesch and Van Deusen 2010). Nguyen et al. (2021) employ a long short-term memory (LSTM) autoencoder to detect anomalies from the time series of sales quantity. Luo et al. (2020) propose to build a directed correlation network based on the purchase volume time series and use a graph-cut method for cheating detection. Given the aforementioned rich studies, however, deterring the gray market has not been well-resolved with machine learning techniques. In general, this paper aims to make advancements in both the time series analytical techniques and long-standing yet unsolved problem in supply chain management and contribute to both fields. To the best of our knowledge, we are among the first to solve the problem of gray market deterrence from a data-driven perspective.

2.2. Anomaly Detection from Time Series

Identifying anomalies in time series data is a well-established field in statistics and machine learning, and it investigates approaches to learn normality from a set of data with time correlations and detect anomalies that deviate from the normal model (Blázquez-García et al. 2021). Two major types of anomaly detection goals are extensively studied in time series data. The first one is to detect short subsequences or time points as anomalies. The other line of research is to identify the whole time series as an anomaly with respect to a set of other normal time series. Our work aims at identifying whether the TSOQ of a distributor is anomalous or not as a whole and, therefore, falls into the second category of studies.

Popular methods to detect anomalous time series usually involve the extraction of various handcrafted time series characteristics based on experts' domain knowledge. For example, Hyndman et al. (2015) propose to extract some representative features, such as lag correlation, strength of seasonality, and spectral entropy, to characterize a time series and use principal component analysis (PCA) to project them into the first two principal components; the time series anomalies are then identified with the density estimated in the PCA space or with the α-hull method (Pateiro-López and Rodríguez-Casal 2010). An R package called "tsfeatures" (Hyndman et al. 2022) is proposed for the purpose of extracting features from time series data as described. Laptev et al. (2015) also propose to construct several time series features, including trend and seasonality, autocorrelation, and average Euclidean distance, but use clustering to categorize the time series into different groups; abnormality is measured by the deviation between the cluster centroid and time series.

Original raw time series data can also be directly used for the purpose of anomaly detection. This category of methods is mainly based on the dissimilarity calculation of pairwise time series. For example, Benkabou et al. (2018) propose to measure the dissimilarity of time series based on a robust dissimilarity calculation method considering the presence of time shift, that is, dynamic time warping, and pose the time series anomaly detection task as a weighted clustering problem. Beggel et al. (2019) utilize a shapelet learning technique to extract representative subsequences of normal time series and then calculated the Euclidean distance between the learned shapelets and subsequences of a time series as the anomaly score.

Whereas still being a respectable choice for anomaly detection, handcrafted feature engineering and raw time series–based methods might lead to suboptimal performance because of their limitation in characterizing large collections of time series data.

2.3. Time Series Representation Learning

It is deemed challenging to detect anomalous time series because anomalies may only be manifested within a long time period (Beggel et al. 2019). On the other hand, deep learning has raised increasing attention because of its capability in automatically characterizing different types of data, such as graphs (Chen et al. 2022, Tian et al. 2022) and texts (Yang et al. 2022). Along this line, many recent studies have turned to utilizing deep learning techniques to learn representations for time series data. Among these techniques, LSTM and gated recurrent units (GRUs) are found to be particularly effective to process temporal information with long-term time dependence. For example, several time series anomaly detection tasks (Malhotra et al. 2015, Hundman et al. 2018) use LSTMs for future prediction and utilize the errors between the predicted values and actual values as the anomaly scores. Recently, because of the success in sequence-to-sequence learning tasks (Sutskever et al. 2014), an LSTM-based encoder-decoder (namely, sequence autoencoder) method (Malhotra et al. 2016) was employed for anomaly detection, and it treated the reconstruction errors as the anomaly scores and was observed to achieve better anomaly detection performance on unpredictable time series data when compared with vanilla LSTM-based methods (Malhotra et al. 2015).

More recently, there arises a line of studies on learning representations for multivariate time series data by taking account of the interdependence of time variables (Zhou et al. 2018, Zhang et al. 2019). Zhou et al. (2018) propose to solve the anomaly detection task with a nonparametric model, in which a regularization term is introduced to capture interseries relatedness. Zhang et al. (2019) propose a deep neural network–based model called a multiscale convolutional recurrent encoder-decoder (MSCRED) to jointly consider the intersensor (time series) correlations and temporal dependencies. MSCRED uses a convolutional encoder to model intersensor correlations and employs an attention-based convolutional LSTM to model temporal patterns hidden in multiscale signature matrices. Nevertheless, existing studies along this line seldomly model the complex confounding factors among the multiple channels of the time series, and hence, the representations may not be effective to reveal the intrinsic characteristics, and this indeed motivates our work in this paper.

3. Product Diversion Detection via Multivariate Time Series Representation Learning

As discussed previously, in order to discover the hidden gray market with possible product diversion by downstream distributors, manufacturers can directly monitor the consecutive ordering quantities of the distributors. Typically, each distributor orders products from the manufacturer with a contract-based temporal periodicity, for example, ordering products per month or per quarter. Thus, a distributor's historical ordering records can be organized as TSOQ. Moreover, the sales catalog of a manufacturer often contains many different SKUs, each denoting a distinct item for sale. As a result, the TSOQ of a distributor is indeed a multivariate time series with each SKU corresponding to a specific dimension of the time series. Moreover, considering that, in reality, only a few distributors could be labeled as suspicious product diverters through costly financial audits, supervised learning–based anomaly detection may not be applicable for practical use.[1] Therefore, we formulate our problem as detecting product diversion from distributors' TSOQ under an unsupervised setting.

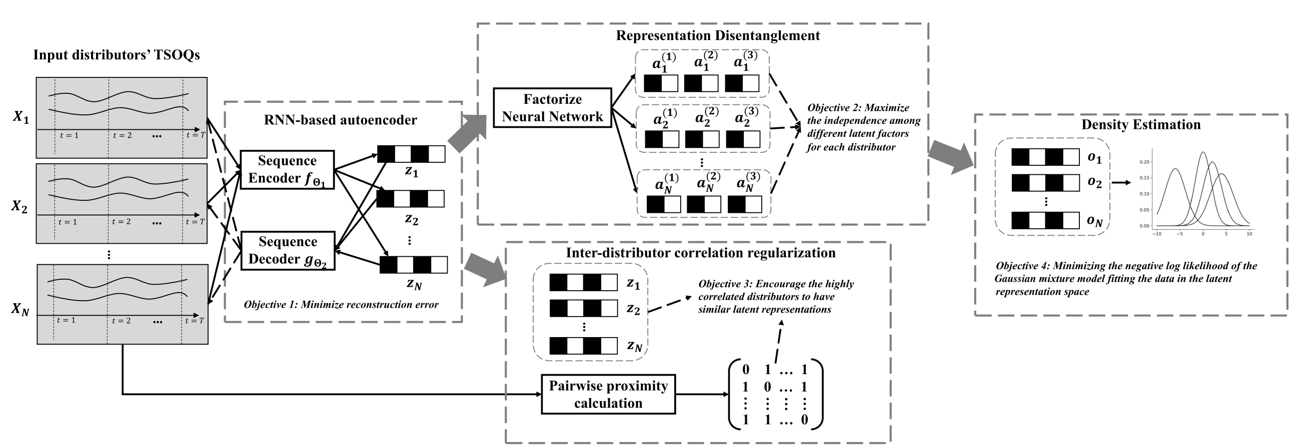
Formally, let $\mathcal{P}$ denote a set of N distributors in the authorized distribution channels of a focal manufacturer, and the ordering quantities of each distributor $p \in \mathcal{P}$ can be organized as a TSOQ, that is, $\boldsymbol{X}_p \in \mathbb{R}_+^{M \times T}$, where T denotes the number of time steps in the ordering history and M represents the number of time-series channels that is exactly the number of SKUs in the sales catalog of the manufacturer. Then, the problem of product diversion detection can be defined as detecting anomalous TSOQ so that a subset of distributors $\mathcal{P}^* \subset \mathcal{P}$ can be identified among which the ordering quantities $\boldsymbol{X}_{p^*}$ of each distributor $p^* \in \mathcal{P}^*$ significantly deviate from the normal ones.[2]

3.1. The Model Framework

Given the previously defined problem, an intuitive solution is to extract informative temporal patterns from the multivariate TSOQ as the behavioral patterns of normal distributors, and the distributors who behave differently from the normal patterns could be regarded as anomalies. This way of thinking, however, poses various challenges that are closely related to the very nature of multivariate time series. First, the temporal patterns of anomalous product diversion behaviors are hidden inside the long time period of TSOQ, which calls for models that can go beyond traditional handcrafted features or limited domain knowledge. Meanwhile, the ordering quantities of different distributors in the same channel could exhibit strong correlations for complex factors, such as product diversion, seasonal effects, etc. Finally, the channels corresponding to different cataloged SKUs in TSOQ are possibly intercorrelated, but the factors that drive the cross-channel correlations are complex and latent as well. In light of these challenges, we resort to deep representation learning–based methods to capture the temporal, cross-channel, and cross-distributor characteristics of the multivariate TSOQ for product diverter detection.

Figure 1 illustrates the framework of our model, which is composed of three main modules, that is, a recurrent neural network (RNN)-based autoencoder with a disentangled learning layer for multivariate TSOQ representation, an interdistributor correlation regularizer, and a density estimator for anomalous score computation. Concretely, we first adopt a sequence autoencoder to derive an overall representation for each TSOQ. To further capture the complex latent factors in cross-channel correlations of the multivariate time series, we develop a disentangled learning layer to decouple a multivariate TSOQ into several latent aspects, each reflecting certain mutual temporal patterns of some coherent SKUs. We also consider the correlations between the distributors when learning the representations of the time series by imposing a regularizer. Finally, given the unsupervised setting of our problem, we conduct density estimation with a GMM in the representation space; the distributors whose representations are estimated with lower density based on the fitted GMM are considered more likely to be anomalies. We develop an end-to-end learning framework to integrate the multiple modeling modules. In this way, we can optimize all the parameters simultaneously. In what follows, we elaborate on the details of each module.[3]

Figure 1. The Framework of the Proposed Model



Note. Three latent factors are assumed to be disentangled.

3.2. RNN-Based Autoencoder

In order to characterize each distributor's TSOQ within a long time period, we employ an RNN-based sequence autoencoder (Sutskever et al. 2014) to reconstruct multivariate time series data. The basic idea of the sequence autoencoder is to first use an encoder to compress the input time series into a low-dimensional embedding vector and then use a decoder to reconstruct the input time series from the vector. We choose a sequence autoencoder instead of a vanilla RNN because the former is reported to show better downstream anomaly detection performances than the latter (Malhotra et al. 2015). Moreover, we choose a deterministic autoencoder rather than a generative autoencoder such as a variational autoencoder because the latter suffers severely from the well-known posterior collapsing issue.

Specifically, given the multivariate TSOQ of a distributor $p \in \{1, 2, \ldots, N\}$, an RNN encoder $f_{\boldsymbol{\Theta}_1}(\cdot)$ is used to map its whole sequence $\boldsymbol{X}_p \in \mathbb{R}_+^{M \times T}$ into a low-dimensional representation vector $\boldsymbol{z}_p \in \mathbb{R}^H$, where H is the dimensionality of the latent representation:

$$\begin{aligned} \boldsymbol{z}_p &= f_{\boldsymbol{\Theta}_1}(\boldsymbol{X}_p), \text{ with} \\ \boldsymbol{X}_p &= [\,\boldsymbol{x}_{p,1} \quad \boldsymbol{x}_{p,2} \quad \cdots \quad \boldsymbol{x}_{p,t} \quad \cdots \quad \boldsymbol{x}_{p,T}\,], \\ \boldsymbol{x}_{p,t} &= [\,x_{p,t,1} \quad x_{p,t,2} \quad \cdots \quad x_{p,t,m} \quad \cdots \quad x_{p,t,M}\,]^\top, \end{aligned} \tag{1}$$

where $\boldsymbol{\Theta}_1$ is the set of neural network parameters of the encoder $f_{\boldsymbol{\Theta}_1}(\cdot)$ and $x_{p,t,m}$ denotes the ordering quantity of the mth SKU at time step t for the distributor p. The latent representation $\boldsymbol{z}_p$ is then fed to an RNN decoder $g_{\boldsymbol{\Theta}_2}(\cdot)$ for generating a new sequence $\hat{\boldsymbol{X}}_p \in \mathbb{R}^{M \times T}$, which is expected to be as close as $\boldsymbol{X}_p$:

$$\hat{\boldsymbol{X}}_p = g_{\boldsymbol{\Theta}_2}(\boldsymbol{z}_p), \tag{2}$$

where $\boldsymbol{\Theta}_2$ denotes a set of neural network parameters for the decoder $g_{\boldsymbol{\Theta}_2}(\cdot)$.[4]

In order to optimize the parameters $\boldsymbol{\Theta}_1$ and $\boldsymbol{\Theta}_2$, we minimize a reconstruction loss that is the mean squared error between $\boldsymbol{X}_p$ and $\hat{\boldsymbol{X}}_p$. Note that we reconstruct the sequence in a reverse order (Sutskever et al. 2014, Malhotra et al. 2016), and thus, the reconstruction loss is defined as

$$\mathcal{L}_1 = \frac{1}{NT} \sum_{p=1}^{N} \sum_{t=1}^{T} \|\boldsymbol{x}_{p,t} - \hat{\boldsymbol{x}}_{p,T-t+1}\|_2^2. \tag{3}$$

3.3. Disentangled Representation Learning for TSOQ

In reality, the TSOQ of one SKU can be correlated with the TSOQ of another SKU because the demands across different SKUs may exhibit latent interdependence because of multiple factors. One typical factor is that the ordered SKUs can be substitutable or complementary, leading to negatively or positively correlated TSOQs. In addition, the original TSOQ may contain noisy information across a vast majority of distributors; for example, some distributors may order low or even zero quantities for some SKUs because of a shortage of funds. Without modeling such interdependence among SKUs and denoising the original time series, the temporal patterns hidden in TSOQ cannot be correctly uncovered. Note that, although the sequence autoencoder has also implicitly modeled the interdependence via the GRU, it cannot disentangle the latent factors for the interdependence, which is prone to learning ineffective temporal patterns as well as entangled correlations in between and may result in suboptimal performances for downstream detection tasks.

Disentangled representation learning (Bengio et al. 2013), aiming to produce factorized representations that disentangle the hidden explanatory factors behind the observed data, has gained increasing attention in recent years. The informativeness and robustness of the factorized representations find many promising applications in product recommendation (Ma et al. 2020), graph representation learning (Li et al. 2021), computer vision (Sanchez et al. 2020), etc. Therefore, to overcome the limitations of the sequence autoencoder, we propose a sequence disentanglement module as an essential complement to the sequence reconstruction procedure.

3.3.1. Factorizing Multivariate Sequences.

The first step of the sequence disentanglement module is to transform the representation $\boldsymbol{z}_p$ yielded from the sequence autoencoder into a factorized representation $\boldsymbol{A}_p = [\,\boldsymbol{a}_p^{(1)} \quad \boldsymbol{a}_p^{(2)} \quad \cdots \quad \boldsymbol{a}_p^{(G)}\,] \in \mathbb{R}^{H \times G}$ for each distributor p, where G is the total number of latent factors. The gth component $\boldsymbol{a}_p^{(g)} \in \mathbb{R}^H$ corresponds to the representation of distributor p with respect to the gth latent factor, for example, product category or distributor intention. We have

$$\boldsymbol{A}_p = F_{\boldsymbol{\Theta}_3}(\boldsymbol{z}_p), \tag{4}$$

where $F_{\mathbf{\Theta}_3}(\cdot)$ denotes a neural network for the factorizing purpose, that is, a factorizing network, with a set of parameters $\mathbf{\Theta}_3$.

Specifically, the transformation starts by calculating a weight $w_p^{(g)} \in \mathbb{R}$ that measures how much information of the sequence reconstruction representation $\boldsymbol{z}_p$ contributes to $\boldsymbol{a}_p^{(g)}$:

$$w_p^{(g)} = \frac{\exp\left\{\frac{1}{\sqrt{H}} LN_1(\boldsymbol{z}_p)^\top LN_2(\boldsymbol{c}_g)\right\}}{\sum_{j=1}^{G} \exp\left\{\frac{1}{\sqrt{H}} LN_1(\boldsymbol{z}_p)^\top LN_2(\boldsymbol{c}_j)\right\}}, \tag{5}$$

where $g \in \{1, 2, \ldots, G\}$. Here, $\boldsymbol{c}_g \in \mathbb{R}^H$ is a learnable parameter that indirectly controls the impact of $\boldsymbol{z}_p$ over $w_p^{(g)}$. Let $LN_l(\cdot)$ denotes a layer-normalization layer (Ba et al. 2016) with a learnable weight vector parameter $\boldsymbol{w}_l \in \mathbb{R}^H$ and a learnable bias vector parameter $\boldsymbol{b}_l \in \mathbb{R}^H$. The layer-normalization layer enables input data normalization by calculating the mean and variance in order to stabilize the neural network training. The subscript l indicates different layer-normalization layers. We can then compute the gth component of the factorized representation as follows:

$$\boldsymbol{a}_p^{(g)} = LN_3(w_p^{(g)} \boldsymbol{z}_p + \boldsymbol{b}^{(g)}), \tag{6}$$

where $\boldsymbol{b}^{(g)} \in \mathbb{R}^H$ is a learnable bias vector parameter and we initialize it by sampling from a normal distribution with a zero mean and a standard deviation of $1/\sqrt{H}$. Overall, the learnable parameter set $\mathbf{\Theta}_3$ in the factorizing network can be represented as $\{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_G, \boldsymbol{w}_1, \boldsymbol{b}_1, \boldsymbol{w}_2, \boldsymbol{b}_2, \boldsymbol{w}_3, \boldsymbol{b}_3, \boldsymbol{b}^{(1)}, \ldots, \boldsymbol{b}^{(G)}\}$. In this way, we obtain the factorized latent factors for each sequence representation.

3.3.2. Disentangling Latent Factors. In order to disentangle the latent factors, a natural way is to minimize the statistical dependency across the stochastic variables that generate these factors. Hence, we need to first measure the statistical dependency across these variables. A useful tool to achieve this purpose is total correlation (Watanabe 1960), which is defined as the Kullback–Leibler (KL) divergence between the variables' joint density and the product of their marginal densities. Therefore, the disentanglement learning with the aim of minimizing the statistical dependency between the latent factors can be regarded as minimizing the total correlation.

However, exact KL divergence minimization is computationally intractable because KL divergence is difficult to compute when we do not have the densities of these variables' joint or marginal distributions and only have access to their samples. In order to address this challenge, we have to achieve the KL divergence minimization in an approximate way. In essence, KL divergence minimization can be regarded as minimizing the difference between the two distributions, that is, matching two distributions. Having only access to the samples from two distributions instead of their probability densities, generative adversarial networks offer a framework for matching the two distributions (i.e., data and model distributions) via an adversarial training procedure involving two neural networks, including generator and discriminator networks.

Following this intuition, we propose to utilize an adversarial training strategy to match the joint distribution and the product of their marginal distributions, which serves as a proxy to achieve KL divergence minimization. Formally, we assume that, $\forall p \in \{1, 2, \ldots, N\}$, the G latent factors $\boldsymbol{a}_p^{(1)}, \boldsymbol{a}_p^{(2)}, \ldots, \boldsymbol{a}_p^{(G)}$ are observations of G stochastic variables $\{Z_1, Z_2, \ldots, Z_G\}$, respectively. In this way, $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_N$ can be treated as N samples from the joint distribution $p(Z_1, Z_2, \ldots, Z_G)$. Using the standard resampling trick in the independence testing literature (Arcones and Gine 1992, Brakel and Bengio 2017, Kim and Mnih 2018, Sanchez et al. 2020), we can obtain N samples that are assumed to be generated from the product of marginal distributions, that is, $p(Z_1)p(Z_2)\cdots p(Z_G)$, by randomly permuting across the batch for each latent dimension of $\{\boldsymbol{A}_p\}_{p=1}^N$. We denote the permuted samples as $\hat{\boldsymbol{A}}_1, \hat{\boldsymbol{A}}_2, \ldots, \hat{\boldsymbol{A}}_N$. Then, a discriminator network $D_{\mathbf{\Theta}_4}(\cdot) : \mathbb{R}^{H \times G} \to \mathbb{R}$ with a set of parameters $\mathbf{\Theta}_4$ is designed to produce the probability that the input is a sample from $p(Z_1, Z_2, \ldots, Z_G)$ rather than from $p(Z_1)p(Z_2)\cdots p(Z_G)$. In order to optimize $\mathbf{\Theta}_4$, we maximize the following objective function:

$$\mathcal{L}_2 = \frac{\lambda_{dis}}{N}\left[\sum_{p=1}^{N} \log D_{\mathbf{\Theta}_4}(\boldsymbol{A}_p) + \sum_{q=1}^{N} \log(1 - D_{\mathbf{\Theta}_4}(\hat{\boldsymbol{A}}_q))\right], \tag{7}$$

where λ_{dis} is a hyperparameter that controls the impact of adversarial training on the overall objective function. After optimizing $\mathbf{\Theta}_4$, we optimize other parameters by minimizing the losses of other modules and $\mathcal{L}_2$. The full optimization process can be viewed as a min-max game. We present more detailed derivation of the adversarial training process in Online Appendix A.2.[5]

3.4. Interdistributor Correlation Regularization

Whereas different distributors order products from a manufacturer according to their own expected demands, some of them may have similar ordering patterns. For example, two geographically close distributors might probably reduce ordering quantities synchronously for the upcoming off-season sales in the same region. Thus, in constructing the representations for the distributors, we need to consider the correlations between the distributors and encourage the highly correlated distributors to have similar representations.

Along this line, we first construct the relations between every two distributors according to the correlations of their historical ordering quantities in TSOQ. Here, we employ Pearson correlation to derive the proximity between the distributors, which is defined as

$$Proximity_{p,q} = \begin{cases} 0, & corr(\boldsymbol{X}_p, \boldsymbol{X}_q) < \eta \\ 1, & corr(\boldsymbol{X}_p, \boldsymbol{X}_q) \geq \eta \end{cases}, \tag{8}$$

where $corr(\boldsymbol{X}_p, \boldsymbol{X}_q)$ denotes the Pearson correlation between distributors p and q in the input time series space and the threshold η is a manually predefined hyperparameter controlling the strength of correlations between the distributors.

To preserve the correlations of all the distributors in the representation space, we impose a correlation regularization term on Equation (3) and minimize the following loss:

$$\mathcal{L}_3 = \lambda_c \sum_{p=1}^{N} \sum_{q=1}^{N} Proximity_{p,q} \|\boldsymbol{z}_p - \boldsymbol{z}_q\|_2^2, \tag{9}$$

where λ_c is a hyperparameter that controls the influence of interdistributor correlation regularization to the overall objective function.

3.5. Density Estimation for Anomaly Detection

From a data distribution perspective, most normal data are assumed to reside in dense areas, whereas anomalies are separated from the normal and fall into low-density areas (Zhai et al. 2016, Zong et al. 2018). As a result, density estimation is crucially important for unsupervised anomaly detection as in our case. However, direct density estimation over high-dimensional raw TSOQ data without supervision remains a challenging task; if we further take into account the strong randomness in distributors' ordering quantities along with the varying market conditions, the challenge is even greater. So we resort to the lower dimensional representations of the TSOQ and try to build a high-capacity and robust density estimation model.

Deep generative models show great capacities in capturing complex data distributions via deep neural networks. In light of this, given the disentangled representations of the TSOQ, we construct a density estimation module by using the deep Gaussian mixture model (Zong et al. 2018) to characterize the distributions of the temporal patterns of ordering quantities. More formally, given the G disentangled factorized representations $\{\boldsymbol{a}_p^{(g)}\}_{g=1}^{G}$ for each distributor p, we first obtain its overall sequence representation $\tilde{\boldsymbol{z}}_p \in \mathbb{R}^H$: $\tilde{\boldsymbol{z}}_p = 1/G \sum_{g=1}^{G} \boldsymbol{a}_p^{(g)}$. Then, we compute a reconstruction error $e_p \in \mathbb{R}$ between the input sequence $\boldsymbol{X}_p$ and the reconstructed sequence $\hat{\boldsymbol{X}}_p$ by the sequence autoencoder, which serves as an informative feature to distinguish outliers from the normal (Zong et al. 2018):

$$e_p = \varphi(\boldsymbol{X}_p, \hat{\boldsymbol{X}}_p), \tag{10}$$

where $\varphi(\cdot)$ denotes a function for computing the distance between two inputs and is set as cosine distance in our case. The final sequence representation of a TSOQ for density estimation is a concatenated H'-dimensional vector $\boldsymbol{o}_p$ as follows:

$$\boldsymbol{o}_p = [\tilde{\boldsymbol{z}}_p; e_p], \tag{11}$$

where the dimensionality is $H' = H + 1$.

The density estimation module further assumes that distributors in the low-dimensional space, that is, $\{\boldsymbol{o}_p\}_{p=1}^{N}$, are generated from a mixture of K Gaussian distributions, each distribution $k \in \{1, 2, \ldots, K\}$ being associated with a mean parameter $\boldsymbol{\mu}_k \in \mathbb{R}^{H'}$ and a covariance matrix parameter $\boldsymbol{\Sigma}_k \in \mathbb{R}^{H' \times H'}$. In other words, the likelihood for observing $\boldsymbol{o}_p$ is computed as a weighted sum of the probabilities that $\boldsymbol{o}_p$ comes from the K different Gaussian distributions:

$$\mathbb{P}(\boldsymbol{o}_p) = \sum_{k=1}^{K} \phi_k \mathcal{N}(\boldsymbol{o}_p \,|\, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{12}$$

where ϕ_k denotes a prior probability that a data point in $\{\boldsymbol{o}_p\}_{p=1}^{N}$ comes from the kth Gaussian distribution.

A common way to estimate the unknown parameters of the GMM model, that is, $\{\phi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$, given the data $\{\boldsymbol{o}_p\}_{p=1}^N$, involves a two-step training approach by iteratively conducting sequence representation learning and density estimation. However, this approach raises concerns about potential loss of vital information essential for accurate density estimation during the sequence representation learning. To address this limitation, we introduce a neural network for parameter estimation of GMM. This incorporation enables the joint optimization for both parameters of GMM and sequence representation, thereby fostering a cohesive end-to-end training paradigm.

Specifically, we implement a multilayer perceptron layer $E_{\boldsymbol{\Theta}_5}(\cdot): \mathbb{R}^{H'} \to \mathbb{R}^K$ with a softmax activation function included to map $\boldsymbol{o}_p$ to a probability vector $\boldsymbol{\gamma}_p = [\gamma_{p,1} \quad \gamma_{p,2} \quad \cdots \quad \gamma_{p,k} \quad \cdots \quad \gamma_{p,K}]^\top$, where $\gamma_{p,k}$ denotes the probability of $\boldsymbol{o}_p$ belonging to the kth Gaussian in GMM:

$$\boldsymbol{\gamma}_p = E_{\boldsymbol{\Theta}_5}(\boldsymbol{o}_p), \tag{13}$$

where $\boldsymbol{\Theta}_5$ is the set of parameters for $E_{\boldsymbol{\Theta}_5}(\cdot)$. Based on Equation (13), we can calculate the GMM parameters as follows:

$$\phi_k = \sum_{p=1}^N \frac{\gamma_{p,k}}{N}, \quad \boldsymbol{\mu}_k = \frac{\sum_{p=1}^N \gamma_{p,k}\boldsymbol{o}_p}{\sum_{p=1}^N \gamma_{p,k}}, \quad \boldsymbol{\Sigma}_k = \frac{\sum_{p=1}^N \gamma_{p,k}(\boldsymbol{o}_p - \boldsymbol{\mu}_k)(\boldsymbol{o}_p - \boldsymbol{\mu}_k)^\top}{\sum_{p=1}^N \gamma_{p,k}}. \tag{14}$$

Based on Equations (12) and (14), we adopt the negative log-likelihood for observing $\boldsymbol{o}_p$ of each p as the loss function of the deep Gaussian mixture model, which is given by

$$NLL(\boldsymbol{o}_p) = -\log\left[\sum_{k=1}^K \phi_k \frac{\exp\left(-\frac{1}{2}(\boldsymbol{o}_p - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{o}_p - \boldsymbol{\mu}_k)\right)}{\sqrt{|2\pi\boldsymbol{\Sigma}_k|}}\right], \tag{15}$$

where $|\cdot|$ denotes the matrix determinant operation. To avoid the singularity problem in GMM, we further penalize the diagonal entries of each $\boldsymbol{\Sigma}_j$ for being small values. So we minimize the following objective function for density estimation:

$$\mathcal{L}_4 = \frac{\lambda_e}{N}\sum_{p=1}^N NLL(\boldsymbol{o}_p) + \lambda_{\boldsymbol{\Sigma}} \sum_{k=1}^K \sum_{h=1}^{H'} \sum_{h'=1}^{H'} \frac{1}{\Sigma_{k,h,h'}}, \tag{16}$$

where λ_e and $\lambda_{\boldsymbol{\Sigma}}$ are two hyperparameters that control the impacts of the negative log-likelihood term and the diagonal entry penalization term to the overall objective function, respectively.

3.6. Learning Algorithm and Anomaly Score

Given the three modules of our model, that is, the sequence autoencoder with disentangled representation learning, the interdistributor correlation regularization, and the deep GMM density estimation, the overall objective function for the whole model can be defined as a weighted sum of the three modules' objectives:

$$\begin{aligned}\min_{f_{\boldsymbol{\Theta}_1}, g_{\boldsymbol{\Theta}_2}, F_{\boldsymbol{\Theta}_3}, E_{\boldsymbol{\Theta}_5}} \max_{D_{\boldsymbol{\Theta}_4}} \mathcal{L} = &\frac{1}{NT}\sum_{p=1}^N \sum_{t=1}^T \|\boldsymbol{x}_{p,t} - \hat{\boldsymbol{x}}_{p,T-t+1}\|_2^2 + \frac{\lambda_{dis}}{N}\left[\sum_{p=1}^N \log D_{\boldsymbol{\Theta}_4}(\boldsymbol{A}_p) + \sum_{q=1}^N \log(1 - D_{\boldsymbol{\Theta}_4}(\hat{\boldsymbol{A}}_q))\right] \\ &+ \lambda_c \sum_{p=1}^N \sum_{q=1}^N Proximity_{p,q}\|\boldsymbol{z}_p - \boldsymbol{z}_q\|_2^2 - \frac{\lambda_e}{N}\sum_{p=1}^N \log\left[\sum_{k=1}^K \phi_k \frac{\exp\left(-\frac{1}{2}(\boldsymbol{o}_p - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{o}_p - \boldsymbol{\mu}_k)\right)}{\sqrt{|2\pi\boldsymbol{\Sigma}_k|}}\right] \\ &+ \lambda_{\boldsymbol{\Sigma}} \sum_{k=1}^K \sum_{h=1}^{H'} \sum_{h'=1}^{H'} \frac{1}{\Sigma_{k,h,h'}}.\end{aligned} \tag{17}$$

We propose a joint learning algorithm to infer all the parameters in an end-to-end manner. The complete learning algorithm is presented in Online Appendix A.4. With the parameters $\phi, \boldsymbol{\mu}, \boldsymbol{\Sigma}$ learned on the training set, we can calculate the negative log-likelihood $NLL(\boldsymbol{o}_{p'})$ by Equation (15) for each distributor p' in the test set, which serves as the anomaly score for p'. Intuitively, with the GMM parameters estimated with normal distributors in the training data, the log-likelihood for observing anomalies in the testing data should be small. Therefore, the distributors with suspicious product diversion activities should have larger negative log-likelihood values than the normal ones.

4. Evaluations

In this section, we conduct extensive experiments on real-world distributors' historical ordering data to demonstrate the effectiveness of the proposed model. We further validate the model robustness on synthetic data sets under different settings.

4.1. Empirical Data

We collect a real-life distribution channel data set from a world-renowned IT company, in which a total of 896 unique SKUs are listed in the catalog and have been ordered by 4,063 nationwide distributors within a four-year period. The orders are launched by the distributors every month. As a result, each distributor's ordering quantities form a TSOQ that is 896-dimensional and has 48 time steps. The manufacturer has discovered the distributors with suspicious product diversion activities through internal audits and labeled them as anomalous distributors. These anomalies take up only 1.7% of all the distributors. In addition, according to the internal audit reports, the anomalous distributors usually inflate their orders for high-level price discounts and further divert the excess inventories to other authorized distributors at prices in between the procurement prices with high- and low-level discounts. Some samples of TSOQs are presented in Online Appendix C.

4.2. Experimental Setup

Contrary to our proposed method, rarely have prior methods adopted an end-to-end learning framework for the anomaly detection task. Instead, they often adopt a two-stage strategy by first representing TSOQ through feature extraction of sequence learning models and then applying different anomaly detection methods, such as one-class support vector machines (OCSVM) and GMM, to derive the anomaly scores. Concretely, we adopt seven competitive baselines and tailor them for the product diversion detection task to validate the effectiveness of the method.

Tsfeatures–Gaussian kernel density estimation (GKDE) is a time series anomaly detection method by extracting multiple time series features (Hyndman et al. 2015). It extracts several features from time series, such as the lag correlation and spectral entropy, and uses principal component decomposition to obtain their first two principal components. Then, GKDE is used to produce an anomaly score for each time series.

Time-series representation learning framework via temporal and contextual contrasting (TSTCC)–OCSVM is an unsupervised time series representation learning method (Eldele et al. 2021). It constructs two different views of the raw data, that is, weak and strong augmentations, with novel temporal and contextual contrasting modules to learn time series representations from unlabeled data. Then, based on the learned representations, we utilize OCSVM for anomaly detection.

Variational recurrent autoencoder (VRAE) adopts a variational autoencoder with an encoder and decoder parameterized with bidirectional LSTMs to learn informative representations of time series. Wasserstein similarity is exploited to compute the anomaly score for each distributor in the experiment (Pereira and Silveira 2019).

Sequence autoencoder (SEQAE)–Gaussian is an implementation of a seminal work that utilizes a sequence autoencoder for time series anomaly detection (Malhotra et al. 2016). It first trains a sequence autoencoder to reconstruct TSOQ. The errors of reconstruction are then assumed to follow a Gaussian distribution. Based on the learned distribution in the training set, a testing instance with low likelihood is deemed anomalous.

SEQAE-OCSVM is a state-of-the-art anomaly detection method in supply chain management (Nguyen et al. 2021), which utilizes a sequence autoencoder to learn representations and OCSVM for anomaly detection. The sequence autoencoder is built with GRU layers, and the OCSVM is built with the radial basis function kernel.

SEQAE-GMM is a competitive method that we build for time series anomaly detection. To tackle the limited ability of SEQAE-Gaussian in using Gaussian distribution to detect anomalies, we utilize a Gaussian mixture model for anomaly detection, which arises to SEQAE-GMM. It separately trains a sequence autoencoder for TSOQ reconstruction and a feed-forward neural network for GMM-based density estimation.

Cut-Rank is a recent cheating detection method in supply chain management, and it detects cheating behaviors in distribution channels based on the construction of a correlation network and a graph-cut method (Luo et al. 2020).

For the train–validation–test split, we randomly divide the distributor set with a ratio of 1:1:1 to form training, validation, and test sets, respectively. To ensure fair comparison among all competitors, we train each model on the training set, tune the hyperparameters on the validation set for the best performance, and report the detection performances on the test set. Considering the large hyperparameter space, a widely used hyperparameter optimization method, that is, random search (Bergstra and Bengio 2012),[6] is adopted for tuning the hyperparameters, and the details about the tuning process and the best settings[7] of each method's hyperparameters are reported in Online Appendix E.

We adopt two widely used ranking-based metrics for performance evaluation, that is, AUROC and AUPRC. For each metric, a larger value means anomalies are ranked higher in the predicted ranking list, thus indicating better anomaly detection performance. More details about the two metrics are given in Online Appendix B. We implement our model in Pytorch and carry out all the comparison experiments on a Linux workstation with 128 GB memory and 4 GeForce GTX Titan 1,080 Ti GPUs. Each experiment is conducted through 10 runs with the same training and test sets but different parametric initializations. As a result, the reported average performances and the corresponding standard deviations demonstrate the impact of the stochasticity stemming from the random initialization of parameters. Please refer to Lin et al. (2023) for the data, codes, and results of the experiments.

4.3. The Experimental Results

4.3.1. General Performance. The detection performances of different methods on the empirical data set are shown in Table 1, in which the best results are highlighted in bold and the runner-ups are in italics. It can be seen that our method outperforms all the baselines consistently in terms of AUROC and AUPRC, and the improvements are more pronounced in AUPRC. For instance, our method outperforms the second best baseline TSTCC-OCSVM by 64.7% in terms of AUPRC. We also conduct a *t*-test, which shows that the outperformance of our method over the baselines is statistically significant with $p < 0.001$. These indicate that our proposed multivariate time series learning–based method indeed works excellently for product diversion detection.

We can also observe that all time series representation learning–based methods, including TSTCC-OCSVM, VRAE, SEQAE-Gaussian, SEQAE-OCSVM, and SEQAE-GMM, achieve better performances than Cut-Rank and Tsfeatures-GKDE, which are based on raw time series features or handcrafted ones. This implies that time series representation learning indeed excels in extracting high-level demand patterns of a distributor from a whole time series, which is crucial for accurately separating the anomalous distributors from the normal. Moreover, from the perspective of density estimation, we can see that the deep generative model–based method SEQAE-GMM generally performs better than traditional density estimation–based methods, such as SEQAE-OCSVM and SEQAE-Gaussian. This validates the effectiveness of deep generative models in density estimation and product diversion detection. Moreover, considering the potential evaluation biases induced by one time train–test splitting, we also check the robustness of our proposed method by using fivefold cross-validation in Online Appendix H.1.

4.3.2. Ablation Study. Here, we conduct ablation experiments to validate the key design choices of our model, that is, the disentangled learning and interdistributor correlation regularization. To explore the contribution of disentangled learning, we first remove the disentangled learning module, and construct a variant model without disentanglement. To study the impact of interdistributor correlation regularization, we eliminate the correlation regularization term and construct another variant without correlation. We also study the performance of the model with both the disentanglement and interdistributor correlation eliminated and construct a third variant without disentanglement and correlation.

The results of the ablation study are shown in Table 2. It is obvious that the detection performance especially in terms of AUPRC drops significantly when the interdistributor correlation regularization term is removed from the model. Furthermore, by comparing without disentanglement and the full model, we can see that the detection performance also drops sharply after removing the disentangled learning module. These results demonstrate that interdistributor correlation regularization and disentangled learning are both crucial for the success of product diversion detection. Because the AUPRC results of different methods exhibit similar trends with their AUROC results, we only give AUROC results to showcase the detection performances of different methods in the following sections.

Table 1. The Detection Performances of Different Methods on the Empirical Data Set

Method	AUROC	AUROC Improvment, %	AUPRC	AUPRC Improvment, %
Tsfeatures-GKDE	0.4728 (0.0000)	100.1**	0.0181 (0.0000)	1,811.6**
TSTCC-OCSVM	0.8514 (0.0715)	11.1**	*0.2101* (0.0565)	64.7**
VRAE	0.9139 (0.0146)	3.5**	0.1825 (0.0392)	89.59**
Cut-Rank	0.5984 (0.0000)	58.1**	0.0219 (0.0000)	1,521.0**
SEQAE-Gaussian	0.8986 (0.0023)	5.3**	0.0880 (0.0026)	293.2**
SEQAE-OCSVM	0.9080 (0.0172)	4.2**	0.1234 (0.0189)	180.4**
SEQAE-GMM	*0.9191* (0.0086)	2.9**	0.1580 (0.0412)	119.0**
Our method	**0.9460** (0.0117)	n/a	**0.3460** (0.0780)	n/a

Notes. The best results are highlighted in bold and the runner-ups are in italics. The standard deviations for 10 runs are reported in brackets. *t*-test are conducted for the improvements.
*$p < 0.05$; **$p < 0.001$.

Table 2. Ablation Study on the Empirical Data Set

Method	AUROC	AUPRC
Without disentanglement and correlation	0.9029 (0.0165)	0.3210 (0.0692)
Without correlation	0.9411 (0.0092)	0.2377 (0.0775)
Without disentanglement	0.9283 (0.0098)	0.3359 (0.0778)
Full model	**0.9460** (0.0117)	**0.3460** (0.0780)

Note. The best results are highlighted in bold.

4.3.3. Visualization of the Learned Representations. One straightforward way to evaluate the quality of the learned representations is to conduct a visualization analysis. We employ *t*-distributed stochastic neighbor embedding (t-SNE) on the empirical data set to map high-dimensional data into a two-dimensional space and visualize them with scatterplots in Figure 2. Specifically, for comparison purposes, we conduct t-SNE on the raw time series and the learned sequence representations of two variants and our model, respectively. Note that, because each raw time series $\boldsymbol{X}_p \in \mathbb{R}_+^{M \times T}$ is a two-dimensional multivariate matrix, we average it over its second dimension before feeding it to t-SNE. The ground-truth anomalies are highlighted with big circles, whereas the normal ones are marked with small dots.

As shown in Figure 2(a), for raw time series instances, t-SNE mixes anomalies with the normal ones completely, which indicates the necessity of using representation learning. Whereas better, the two deep variants, that is, without disentanglement and correlation and without disentanglement as shown in Figure 2, (b) and (c), respectively, also fail to distinguish anomalies from the normal. In contrast, our proposed method pushes anomalies to some low-density regions, for example, edges of the visualization in Figure 2(d), making clear separations between the anomalies and normal ones. This validates that both modules are of great help to the product diversion detection task.

4.4. Evaluation of the Economic Value with Cost–Benefit Analysis

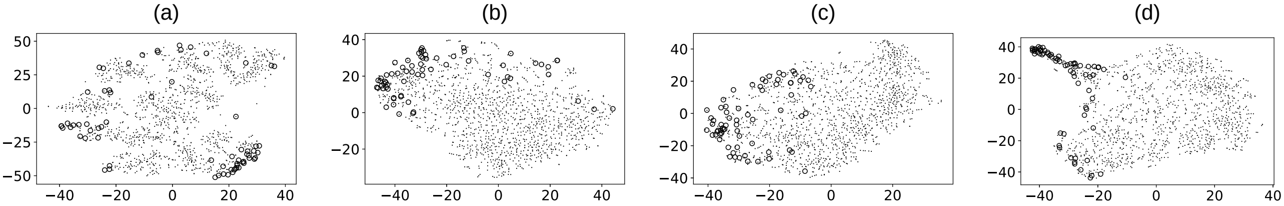
In order to evaluate the economic value of our proposed method, we conduct a cost–benefit analysis. Motivated by prior studies (Provost and Fawcett 2013), the economic value (EV) of a predictive model is computed as follows:

$$EV(Model) = Pr(P) \cdot [TPR \cdot BC_{TP} + FNR \cdot BC_{FN}] + Pr(N) \cdot [FPR \cdot BC_{FP} + TNR \cdot BC_{TN}], \quad (18)$$

where $Pr(P)$ and $Pr(N)$ are the ratios of the number of positive (anomalous) and negative (normal) distributors to the total number of distributors in the ground-truth data, respectively. TPR denotes the true positive rate calculated as $TPR = TP/(TP + FN)$. FNR is the false negative rate calculated as $FNR = FN/(TP + FN)$. FPR is the false positive rate calculated as $FPR = FP/(FP + TN)$. TNR is the true negative rate calculated as $TNR = TN/(TN + FP)$. BC represents the corresponding benefit/cost value.

Because the true cost and benefit for the product diversion detection varies from one to another, we estimate the costs and benefits on average. First, we assume an anomalous distributor uses 50% of total ordering quantities for product diversion by reselling them to other distributors and improperly enjoys the quantity discount. For simplicity, we assume the manufacturer offers the same price discount rate r to every unit purchased once the quantity discount is qualified. For illustration, we set $r = 60\%$ in our case, and thus, if a true positive instance is detected, approximately $1 - r$ revenue can be saved because these quantities should not enjoy the discount without overstocking for product diversion. Moreover, in our empirical data set, the average ordering payment per distributor is $344,536. In this way, we can compute the benefit for a true positive instance as $\$114,845.33 = \$344,536 * 0.5/0.6 * (1 - 0.6)$. For such a true positive instance, there is also an investigation cost that needs to be paid. We assume the investigation cost

Figure 2. Visualization of Learned Representations Using t-SNE on the Empirical Data Set



Notes. (a) Raw data. (b) Without disentanglement and correlation. (c) Without disentanglement. (d) Full model.

rate to be 1%; that is, this cost takes up 1% of the average ordering amount per distributor. Therefore, we have $BC_{TP} = \$114,845.33 - \$344,536 * 0.01 = \$111,399.97$. In contrast, a false positive instance detected by the model can indeed raise a false alarm and incur a cost of investigation. However, the benefit is zero. Thus, we have $BC_{FP} = \$0 - \$344,536 * 0.01 = -\$3,445.36$. For false negative instances, we miss the opportunity to catch the fraud but would not initiate an investigation either; thus, we have $BC_{FN} = \$0 - \$0 = \$0$. In addition, true negative instances can bring neither extra benefit nor cost, and $BC_{TN} = \$0 - \$0 = \$0$.

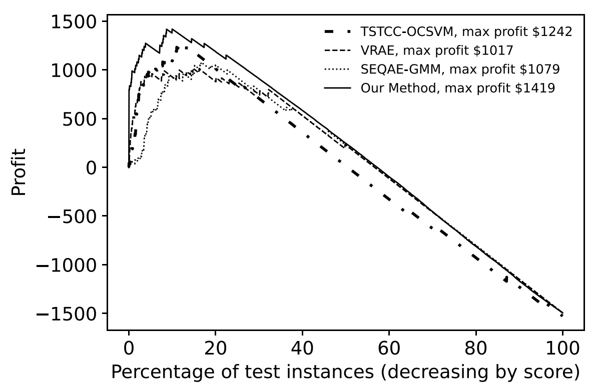
Given the anomaly scores for the testing instances, we need to define a threshold to determine the labels with which we can calculate the EV according to Equation (18). Thus, by varying the threshold, we can obtain a profit curve as shown in Figure 3. The profit curve can help to determine the threshold that can produce the largest EV of a predictive model as practical guidance. As can be seen, the profit curve of our method lies above all the baseline methods, indicating that our method is the most cost-effective overall. The maximum profit can be gained when 12% of distributors are predicted as diverters in our method, that is, EV = \$1,419, which is \$177 larger than the second best method TSTCC-OCSVM. Meanwhile, we can see that the profit begins to decline almost linearly for all the methods when more than 40% distributors are predicted as diverters, which is because all the true diverters have already been identified and additional detection can only bring false positive errors with investigation cost. Additionally, our proposed model shows its advantage in achieving a larger EV when the discount rate of the manufacturer is small and more revenues can be saved through more accurate detection. The details of the cost–benefit analysis are given in Online Appendix H.3.

4.5. Robustness Check with Synthetic Data

In order to check the robustness of our model's performance, we also generate synthetic data for more comprehensive experimental validation. Specifically, we utilize a multiagent-based simulation toolkit called Mason (Luke et al. 2005) to artificially generate a virtual market with 9,000 distributors and 896 unique SKUs. These distributors are enforced by Mason to behave like the normal ones of the real-world data set and to order products over a period of 48 time steps, which forms the initial TSOQ. Then, we randomly choose a set of distributors out of the 9,000 distributors and perturb their TSOQ in a pair-wise manner to simulate the product diversion behaviors, which determines the anomaly labels. By this means, we can construct various synthetic data sets with different levels of anomalies for model training and testing, which are valuable supplements to the real-world data set with very limited anomalies. More details about the simulation settings can be found in Online Appendix D.

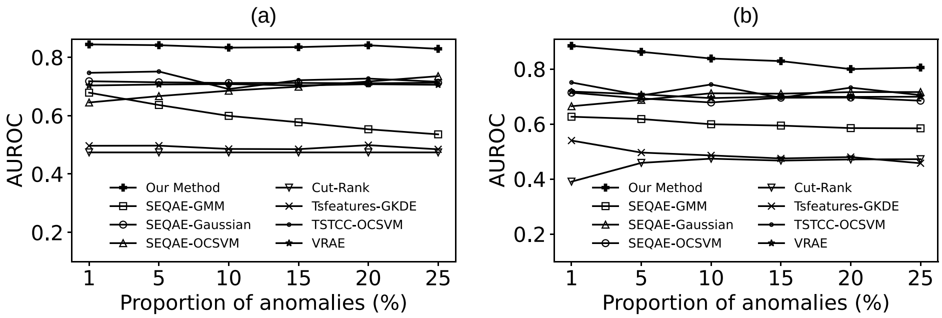
We first study the model performance under different proportions of training anomalies. We randomly split the 9,000 distributors evenly as the training and test sets and generate six synthetic data sets, each of which contains a test set with 10% anomalies and a training set with {1%,5%,10%,15%,20%,25%} anomalies, respectively. We then apply models on the synthetic data sets and evaluate the performances by AUROC. As can be seen in Figure 4(a), our method demonstrates stable performance under different proportions of training anomalies and consistently outperforms all the baseline methods with large margins. This indicates the robustness of our model to different degrees of anomaly pollution in the training data. We also validate the generalization ability of the models by setting different proportions of testing anomalies. Analogously, we generate six synthetic data sets, each of which contains a training set with 10% anomalies and a test set with {1%,5%,10%,15%,20%,25%} anomalies, respectively. As

Figure 3. Performance Comparison with Profit Curves



Note. The *x*-axis denotes the percentage of the distributors predicted as positive (diverters), and the *y*-axis denotes the corresponding EV (profit) of the model.

Figure 4. Detection Performance of Different Methods Under Different Levels of Training or Testing Anomalies



Notes. (a) Different anomaly levels in training. (b) Different anomaly levels in testing.

shown in Figure 4(b), all the methods exhibit stability in detection performance with varying proportions of testing anomalies, and our proposed method consistently outperforms the alternatives, albeit with a slight reduction in the observed performance margin as the quantity of anomalies in the testing data set increases. Furthermore, we also simulate an additional scenario that the anomalous distributors divert their overstocked products to unauthorized merchants in the gray market, which is presented in Online Appendix H.4.

5. Conclusions and Discussions

In this paper, we propose a machine learning–based approach to detect distributors that may be involved in product diversion activities and form a gray market that could do harm to manufacturers. Our method leverages the recent advances in representation learning of time series data to reveal the temporal behavioral patterns of downstream distributors. To identify the complex and latent cross-channel correlations of multivariate time series composed of the ordering records of different SKUs, we propose a disentangled representation learning method by factorizing multivariate time series into several latent factors. Moreover, we impose a regularization constraint on the representations to further capture the correlations among distributors. Empirical experiments on both the real-world and synthetic data sets demonstrate the superiority of the proposed method over state-of-the-art methods, in which the key modeling modules all play vital roles.

Our study has the following managerial implications. First, given the ordering and sales data of the downstream distributors recorded in the existing information systems, the manufacturers can detect suspicious distributors with our proposed anomaly detection method, and they can further launch more targeted and cost-effective internal audits toward the distributors involved in product diversion activities. This cannot only save huge costs in conducting detailed and periodical audits for each distributor, but also help firms maintain a stable market with efficient incentive programs, such as quantity discounts. Additionally, we do not need to keep the traceability of each product in the supply chain by constructing complex and expensive information infrastructures, such as IoTs and blockchains. Conversely, machine learning–based methods provide a more feasible way to probe the product diversion activities of distributors by merely tracking the sales data, which sheds light on new perspectives in monitoring and controlling the gray market for manufacturers.

Second, as one of the pioneering studies that addresses supply chain fraud detection from a data-driven perspective, this work also sheds light on combating other types of supply chain frauds. For example, manufacturers can tackle inventory fraud detection through learning the representations of supply chain data to reduce manual efforts for physical inventory counts. This encourages firms to construct smarter supply chain systems that integrate more heterogeneous data, including retailing, ordering, and environmental data.

Finally, the success of representation learning for TSOQs sheds light on resolving high-dimensional, unstructured data in management decisions via deep representation learning. For example, except for time series, unstructured data, such as images, texts, and other media data, have become pervasive for a company, which can help disclose possible risks it may face. Deep representation learning can also be exploited in this case, in which many pretrained models, such as Bidirectional encoder representations from transformers (BERT) for texts and ImageNet for images, prevail. In particular, our method features decoupling the mixed semantics by creating several disentangled representations, which can be applied to other managerial problems because the data of focal entities in the real world are often entangled with interfering information.

Our research still has several limitations to be addressed in the future. First, the current research only focuses on the time series of ordering quantities with fixed time intervals. However, many real-world time series in supply chains may be noisy with missing values or irregular time intervals. Representing the long time series in such scenarios is generally more challenging, for which new encoders have to be developed. In addition, our method only detects suspicious diversion behaviors at the individual level, whereas product diversion may involve multiple downstream distributors with collusive behaviors, and thus, we can enhance the distributor representations with the distributors' group activities.

Acknowledgments
The authors thank the editors and anonymous referees for their insightful and detailed comments that have significantly improved this paper.

Endnotes

[1] In Online Appendix G.1, we provide more discussions on why we choose to solve the problem in the unsupervised setting.

[2] According to the definition, "anomaly" refers to the distributor that is involved in product diversion, and we use "anomaly" and "product diverter" interchangeably later.

[3] Because each module involves multiple neural network layers, we also illustrate the detailed network structures of these modules in Online Appendix A.3.

[4] More details about the encoder and decoder are available in Online Appendix A.1.

[5] It is also worth noting that there is an alternative learning way to directly estimate and minimize the KL divergence, and it uses a density ratio estimation method. We justify the choice of using the adversarial training process by comparing the detection performances of our proposed model and the density ratio estimation method in Online Appendix G.2.

[6] For efficiency concerns, we suggest applying a python library Ray Tune that can support parallel tuning of hyperparameters in practical applications.

[7] Unless otherwise specified, we use the reported best hyperparameter settings throughout all experiments.

References

Ahmadi R, Yang BR (2000) Parallel imports: Challenges from unauthorized distribution channels. *Marketing Sci.* 19(3):279–294.

Ahmadi R, Iravani F, Mamani H (2015) Coping with gray markets: The impact of market conditions and product characteristics. *Production Oper. Management* 24(5):762–777.

Antia KD, Bergen M, Dutta S (2004) Competing with gray markets. *MIT Sloan Management Rev.* 46(1):63.

Antia KD, Bergen ME, Dutta S, Fisher RJ (2006) How does enforcement deter gray market incidence? *J. Marketing* 70(1):92–106.

Arcones MA, Gine E (1992) On the bootstrap of *U* and *V* statistics. *Ann. Statist.* 20(2):655–674.

Askenaes L, Westelius A (2000) Five roles of an information system: A social constructionist approach to analyzing the use of erp systems. *Proc. 21st Internat. Conf. Information Systems* (Association for Information Systems, Atlanta), 426–434.

Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. Preprint, submitted July 21, https://arxiv.org/abs/1607.06450.

Beggel L, Kausler BX, Schiegg M, Pfeiffer M, Bischl B (2019) Time series anomaly detection based on shapelet learning. *Comput. Statist.* 34(3):945–976.

Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Machine Intelligence* 35(8):1798–1828.

Benkabou S-E, Benabdeslem K, Canitia B (2018) Unsupervised outlier detection for time series by entropy and dynamic time warping. *Knowledge Inform. Systems* 54(2):463–486.

Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. *J. Machine Learn. Res.* 13(2):281–305.

Blázquez-García A, Conde A, Mori U, Lozano JA (2021) A review on outlier/anomaly detection in time series data. *ACM Comput. Surveys* 54(3):1–33.

Brakel P, Bengio Y (2017) Learning independent features with adversarial nets for non-linear ICA. Preprint, submitted October 13, https://arxiv.org/abs/1710.05050.

Cespedes FV, Corey ER, Rangan VK (1988) Gray market: Causes and cures. *Harvard. Bus. Rev.* 66(4):75–82.

Chen X, Liu Y, Zhang C (2022) Distinguishing homophily from peer influence through network representation learning. *INFORMS J. Comput.* 34(4):1958–1969.

Dasu S, Ahmadi R, Carr SM (2012) Gray markets, a product of demand uncertainty and excess inventory. *Production Oper. Management* 21(6):1102–1113.

de Boissieu E, Kondrateva G, Baudier P, Ammi C (2021) The use of blockchain in the luxury industry: Supply chains and the traceability of goods. *J. Enterprise Information Management* 34(5):1318–1338.

Ding L, Hu B, Ke C, Wang T, Chang S (2019) Effects of IoT technology on gray market: An analysis based on traceability system design. *Comput. Indust. Engrg.* 136:80–94.

Eldele E, Ragab M, Chen Z, Wu M, Kwoh CK, Li X, Guan C (2021) Time-series representation learning via temporal and contextual contrasting. Preprint, submitted June 26, https://arxiv.org/abs/2106.14112.

HP (2008) HP reaches settlement in gray market case against Axicom Pc Inc. Accessed March 1, 2022, https://www.hp.com/hpinfo/newsroom/press/2008/080311a.html.

Hundman K, Constantinou V, Laporte C, Colwell I, Soderstrom T (2018) Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. *Proc. 24th ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (Association for Computing Machinery, New York), 387–395.

Hyndman RJ, Wang E, Laptev N (2015) Large-scale unusual time series detection. *2015 IEEE Internat. Conf. Data Mining Workshop* (IEEE, Piscataway, NJ), 1616–1619.

Hyndman R, Kang Y, Montero-Manso P, Talagala T, Wang E, Yang Y, O'Hara-Wild M (2022) tsfeatures: Time series feature extraction. R package version 1.0.2.9000. https://pkg.robjhyndman.com/tsfeatures/.

Kim H, Mnih A (2018) Disentangling by factorising. Jennifer D, Andreas K, eds. *Proc. 35th Internat. Conf. Machine Learn.* (PMLR, New York), 2649–2658.

KPMG (2017) Supply chain fraud. Accessed February 11, 2022, https://assets.kpmg/content/dam/kpmg/be/pdf/Markets/supply-chain-fraud.pdf.

Laptev N, Amizadeh S, Flint I (2015) Generic and scalable framework for automated time-series anomaly detection. *Proc. 21th ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (Association for Computing Machinery, New York), 1939–1947.

Li H, Wang X, Zhang Z, Yuan Z, Li H, Zhu W (2021) Disentangled contrastive learning on graphs. Ranzato M, Beygelzimer A, Dauphin Y, Liang PS, Wortman Vaughan J, eds. *Adv. Neural Inform. Processing Systems*, vol. 34 (Curran Associates, Inc., Red Hook, NY), 21872–21884.

Lin H, Liu G, Wu J, Zhao JL (2023) Deterring the gray market: Product diversion detection via learning disentangled representations of multivariate time series. https://dx.doi.org/10.1287/ijoc.2022.0155.cd, https://github.com/INFORMSJoC/2022.0155.

Luke S, Cioffi-Revilla C, Panait L, Sullivan K, Balan G (2005) Mason: A multiagent simulation environment. *Simulation* 81(7):517–527.

Luo P, Shu K, Wu J, Wan L, Tan Y (2020) Exploring correlation network for cheating detection. *ACM Trans. Intelligent Systems Tech.* 11(1):1–23.

Ma J, Zhou C, Yang H, Cui P, Wang X, Zhu W (2020) Disentangled self-supervision in sequential recommenders. *Proc. 26th ACM SIGKDD Internat. Conf. Knowledge Discovery Data Mining* (Association for Computing Machinery, New York), 483–491.

Malhotra P, Vig L, Shroff G, Agarwal P (2015) Long short term memory networks for anomaly detection in time series. *Proc.*, vol. 89 (Presses Universitaires de Louvain, Louvain-la-Neuve, Belgium), 89–94.

Malhotra P, Ramakrishnan A, Anand G, Vig L, Agarwal P, Shroff G (2016) LSTM-based encoder-decoder for multi-sensor anomaly detection. Preprint, submitted July 1, https://arxiv.org/abs/1607.00148.

Myers MB (1999) Incidents of gray market activity among US exporters: Occurrences, characteristics, and consequences. *J. Internat. Bus. Stud.* 30(1):105–126.

Nguyen H, Tran KP, Thomassey S, Hamad M (2021) Forecasting and anomaly detection approaches using LSTM and LSTM autoencoder techniques with the applications in supply chain management. *Internat. J. Inform. Management* 57:102282.

Pateiro-López B, Rodríguez-Casal A (2010) Generalizing the convex hull of a sample: The R package alphahull. *J. Stat. Software* 34(5):1–28.

Pereira J, Silveira M (2019) Unsupervised representation learning and anomaly detection in ECG sequences. *Internat. J. Data Mining Bioinformatics* 22(4):389–407.

Provost F, Fawcett T (2013) *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking* (O'Reilly Media, Inc., Sebastopol, CA).

Roesch FA, Van Deusen PC (2010) Anomaly detection for analysis of annual inventory data: A quality control approach. *Southern J. Appl. Forestry* 34(3):131–137.

Sanchez EH, Serrurier M, Ortner M (2020) Learning disentangled representations via mutual information estimation. *Eur. Conf. Comput. Vision* (Springer-Verlag, Berlin, Heidelberg), 205–221.

Srivastava A, Mateen A (2020) Supply chain contracts in the presence of gray markets. *Decision Sci.* 51(1):110–147.

Su X, Mukhopadhyay SK (2012) Controlling power retailer's gray activities through contract design. *Production Oper. Management* 21(1):145–160.

Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger K, eds. *Proc. 27th Internat. Conf. Neural Inform. Processing Systems*, vol. 2 (MIT Press, Cambridge, MA), 3104–3112.

Thomassey S (2010) Sales forecasts in clothing industry: The key success factor of the supply chain management. *Internat. J. Production Econom.* 128(2):470–483.

Tian H, Zheng X, Zhao K, Liu MW, Zeng DD (2022) Inductive representation learning on dynamic stock co-movement graphs for stock predictions. *INFORMS J. Comput.* 34(4):1940–1957.

Watanabe S (1960) Information theoretical analysis of multivariate correlation. *IBM J. Res. Development* 4(1):66–82.

Yang Y, Zhang K, Fan Y (2022) Analyzing firm reports for volatility prediction: A knowledge-driven text-embedding approach. *INFORMS J. Comput.* 34(1):522–540.

Zhai S, Cheng Y, Lu W, Zhang Z (2016) Deep structured energy based models for anomaly detection. *Proc. 33rd Internat. Conf. Machine Learn.*, vol. 48 (JMLR.org), 1100–1109.

Zhang J (2016) The benefits of consumer rebates: A strategy for gray market deterrence. *Eur. J. Oper. Res.* 251(2):509–521.

Zhang Z, Feng J (2017) Price of identical product with gray market sales: An analytical model and empirical analysis. *Inform. Systems Res.* 28(2):397–412.

Zhang C, Song D, Chen Y, Feng X, Lumezanu C, Cheng W, Ni J, Zong B, Chen H, Chawla NV (2019) A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. *Proc. 33rd AAAI Conf. Artificial Intelligence and 31st Innovative Appl. Artificial Intelligence Conf. and 9th AAAI Sympos. Educational Adv. Artificial Intelligence*, vol. 33 (AAAI Press, Washington, DC), 1409–1416.

Zhao J, Jiang Z-Z, Sun M (2021) A review on the gray market business using a data-driven approach. *Data Sci. Management* 1(1):38–47.

Zhou Y, Zou H, Arghandeh R, Gu W, Spanos C (2018) Non-parametric outliers detection in multiple time series a case study: Power grid data analysis. *Proc. 32nd AAAI Conf. Artificial Intelligence and 30th Innovative Appl. Artificial Intelligence Conf. and 8th AAAI Sympos. Educational Adv. Artificial Intelligence*, vol. 32 (AAAI Press, Washington, DC).

Zong B, Song Q, Min MR, Cheng W, Lumezanu C, Cho D, Chen H (2018) Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. *Sixth Internat. Conf. Learn. Representations Conf. Track Proc.* (Curran Associates Inc., Red Hook, NY), 3072–3090.