IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 16, 2021, accepted April 23, 2021, date of publication May 3, 2021, date of current version May 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077067

Interpretable Stock Anomaly Detection Based on Spatio-Temporal Relation Networks With Genetic Algorithm

MEI-SEE CHEONG[1], MEI-CHEN WU[1], AND SZU-HAO HUANG[2], (Member, IEEE)
[1]Institute of Information Management, National Chiao Tung University, Hsinchu 30010, Taiwan
[2]Department of Information Management and Finance, National Yang Ming Chiao Tung University, Hsinchu 30010, Taiwan

Corresponding author: Szu-Hao Huang (szuhaohuang@nctu.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Contract MOST110-2622-8-009-014-TM1, Contract MOST109-2221-E-009-139, Contract MOST109-2622-E-009-002-CC2, and Contract MOST109-2218-E-009-015; and in part by the Financial Technology (FinTech) Innovation Research Center, National Yang Ming Chiao Tung University.

ABSTRACT Instability in financial markets represents a considerable risk to investors; examples of instability include a market crash caused by systematic risks and abnormal stock price volatility caused by artificial hype. The early detection of abnormal behavior can help investors adjust their strategy and reduce investment risks. We proposed a spatiotemporal convolutional neural network–based relational network (STCNN-RN) model that can learn the complex correlations between multiple financial time-series data sets, and we used genetic algorithms with a constrained gene to discover the time points for outlier companies by fitting the STCNN-RN model; we used these outlier points to identify abnormal situations. Most research on identifying anomalous patterns has been unable to sufficiently explain the reason for anomalies to investors. We applied an interpretability model to enable investors to understand these anomalous time points in relation to companies and discover the key factors giving rise to the anomalies. The experiment results revealed that the proposed model can be used to model multiple financial time-series data sets and to capture anomalous situations in relevant companies. Because this study explored the discovery of anomaly phenomena in all transaction data and the explanation of these abnormalities, investors can understand a stock market situation holistically.

INDEX TERMS Anomaly detection, genetic algorithm, interpretable model, Relation Network.

I. INTRODUCTION

Although artificial intelligence has proved useful in numerous fields and has been applied to financial problems, technologies for processing financial market data are not as mature as those for processing general text or image data. General scientific data can be modeled and analyzed simply. Text or image data can be analyzed from a purely scientific perspective, and common logic can be used to summarize a set of logical, potential, or inherent rules. Financial data tend to change due to factors relating to the environment, politics, the military, and the media. Because numerous factors can affect financial data, it is difficult to summarize the main factors that cause changes in the financial markets. First, it is necessary to collect raw data with complete and multiple characteristics in a short time, second, deep learning must be used to model the current situation of the financial market, and third, the prediction results must be close to the real price changes; thus, the challenge is monumental. Even though representative mathematical methods or prediction model methods had delivered highly accurate results for less challenging problems, those methods have delivered less accurate results for financial data. Even some promising prediction results have suddenly failed, and such prediction methods seem far less effective than judgments and decisions made by humans using past experience.

The financial markets are highly complex, chaotic, and continuously dynamic environments that exert substantial effects on economies. Therefore, the rise and fall of stock prices substantially affect investor earnings. It is a major and difficult challenge to predict stock market prices or trends. In the early twenty-first century, considerable research

The associate editor coordinating the review of this manuscript and approving it for publication was Kok-Lim Alvin Yau.

has addressed this challenge. Machine learning methods have been used to predict stock prices. Alkhatib *et al.* [1] used a k-nearest neighbor (KNN) algorithm and nonlinear regression to predict stock prices; their experiment results revealed that the prediction results of the KNN algorithm were highly similar to actual stock prices. Fenghua *et al.* [2] employed various economic features in a support vector machine (SVM) to make price predictions. The results indicated that the predictive methods that combine the price features into SVMs have stronger performance. Hegazy *et al.* [3] proposed an algorithm that integrates particle swarm optimization (PSO) and a least-squares SVM (LS-SVM) to predict stock prices. The results revealed that the proposed model had a more favorable prediction accuracy and that the PSO algorithm has potential for optimizing an LS-SVM. Kazem *et al.* [4] proposed a forecasting model based on chaotic mapping, a firefly algorithm, and support vector regression (SVR) to predict stock prices. Compared with related algorithms, the proposed model exhibited the highest performance in terms of two error measures: mean squared error (MSE) and mean absolute percent error (MAPE). The preceding discussion indicates that machine learning techniques exhibit favorable performance in stock market price prediction.

In the past, machine learning methods used human knowledge for feature extraction from data. The difference between machine learning and deep learning is that deep learning uses a multilayer neural network to scrutinize the data and extract relevant characteristics [5]. Numerous deep learning methods have been published. Bao *et al.* [6] proposed a novel deep learning framework in which wavelet transforms (WTs), stacked autoencoders (SAEs), and long short-term memory (LSTM) are combined for stock price forecasting. Their results revealed that the proposed model outperforms other similar models in both predictive accuracy and profitability performance. Hafezi *et al.* [7] proposed a bat-neural network multiagent system (BNNMAS) to predict stock price. The results revealed that the BNNMAS performs accurately and reliably; thus, it can be considered a suitable tool for predicting stock prices, especially over the long term. Khare *et al.* [8] used feedforward neural networks and recurrent neural networks (RNN) to forecast short-term stock prices. Their results indicated that the feedforward multilayer perceptron outperforms LSTM at predicting short-term stock prices. Selvin *et al.* [9] identified the latent dynamics in data by using deep learning architectures. These researchers employed RNN, LSTM, and convolutional neural network (CNN) architectures for price prediction of national stock exchange–listed companies and compared the performance of these architectures. The results revealed that the proposed system is capable of identifying some interrelations within the data. These results highlight that a CNN architecture can be applied to identify changes in trends. The preceding discussion reveals that many studies on deep learning techniques have focused on stock price forecasting.

Some studies have used statistical models or traditional financial methods [10], [11] to predict stock prices and trends. However, we believe that financial market data already contain some of the existing knowledge. Any financial system exhibits interactions between financial commodities. Scholars have published numerous reasoning models for objects or entities; these models may operate between the original data reasoning relations. Among reasoning models, relational networks (RNs) [12] are the most common. The method of Santoro *et al.*, proposed at DeepMind in 2017, involves the use of RNs as a simple plug-and-play module to solve problems that fundamentally hinge on relational reasoning. Most RN studies have focused on pattern recognition and classification tasks. To date, most RN studies have mainly focused on pattern recognition and classification tasks.

In addition to studying the interactions between financial products to help investors understand financial markets, many studies have found that anomaly detection plays a vital role in financial investigations. Detecting anomalies can help investors make investment decisions and can also reduce investment risks. Scholars have applied anomaly detection to crowded scenes [13], pricing data [14], network intrusion [15], hyperspectral images [16], and various other topics. For research on anomaly detection in the financial field, the vast majority of studies have used traditional methods as the main research tool. Typical anomalies can be divided into market anomalies and pricing anomalies in the financial market. ''Market anomalies'' refers to the difference in returns and the contradiction between efficient market assumptions. ''Pricing anomalies'' means that the pricing of something (such as stocks or securities) is different from the pricing predicted by the model; the two most representative models are the capital asset pricing model (CAPM) [17] and the Fama–French three-factor model [18].The CAPM and Fama–French three-factor model are linear models of traditional finance. Compared with existing deep learning models, these two models do not completely fit the relevant theories. The complexity inherent in existing deep learning models enables them to simulate these theories well, indicating that existing complex neural network models have a certain generalization ability.Researchers have also applied deep learning techniques to financial market data [19], [20]for anomaly detection. Most research on anomaly detection has not been able to explain the reasons for anomalies; most research has not effectively reminded investors of crucial details. The frequency of anomalous events is irregular, and no clear definition of anomalies can be found in financial data. Financial market data do not have exact labels, making them dissimilar to text or image data when used in supervised learning; this complicates the use of traditional neural networks for training, detection, and response.

In the context of the aforementioned shortcomings and concerns, this paper presents methods for learning stock market trends and capturing the anomalous time points relevant to companies. It is vital to explain and to analyze the causes of anomalies. However, financial deep learning is currently limited to learning the interactive relationship between two sequences. Even though investors seek to comprehend the trends of entire financial markets, the existing models simply cannot meet this challenge. However, research on applying RNs to the visual question answering (VQA) framework has been quite successful. Such systems perform image recognition tasks accurately, and the results for some data sets are superior to human judgments. Given that the input data applied to the learning task of the RN are mostly image data and a RN model can perform object reasoning, we propose a spatiotemporal convolutional neural network–based relational network (STCNN-RN), which can simultaneously process spatiotemporal data and consider the complex interactive relationships of the input data. Initially, we use RN to construct our model, and when our model successfully fits the training data during a training process, we must discover the most striking outlier time points that cannot fit the model. Such discovery is a complex optimization problem, so we use a genetic algorithm with a constrained gene (GACG) to discover the most notable outliers among the anomalous time points. Consequently, we contend that when the model fits most of the training data, the time points that cannot fit the model are anomalous time points that are incompatible with the entire financial market. This fundamentally describes our strategy to discover anomalies. To enable investors to understand these anomalous time points, we use the local interpretable model–agnostic explanations (LIME) interpretable model to discover the key factors of abnormalities.

This study proposes an effective anomaly detection system. This system includes an improved RN to learn the performance levels and interactions of various companies in the stock market, the results of which can assist investors to comprehend the anomalous time points exhibited by companies. Furthermore, investors can be informed of the causes of anomalies, and investors can improve their understanding of anomaly patterns. The research framework that we have established can be used to conduct arbitrage. Despite numerous publications on anomaly detection and analytical methods, but most of which has not addressed time series data, only a few studies have focused on financial data, such as accounting data [21], [22] and credit card data [19]. Unlike these approaches, we propose a novel system that can capture the relationships between multiple companies and examine anomalous trends by using time series data and analyzing the causes of anomaly patterns.

Our major contributions of this research can be summarized as follows:

- RN-based market model: Our proposed research framework is the first to use the STCNN-RN to model the complex cross-correlations between various companies in the stock markets; this framework can fit regular market behaviors between each pair of companies.
- Genetic algorithm–based anomaly detection method: Our proposed STCNN-RN and GACG can discover anomalous time points relevant to companies and can discover major events in stock markets.
- LIME-based interpretable model: We use the LIME interpretable model to analyze and explain the causes of anomalies. Using interpretable models to explain anomaly patterns makes it easier for investors to understand the market situation.
- Practical applications in various financial markets: We validated our studies by using multiple data sets. Minute price data of S&P 100 constituent stocks were taken from Wharton Research Data Services Trade and Quot-e (WRDS TQA) [23]. Minute price data of FTSE TWSE Taiwan 50 index constituent stocks were taken from the Taiwan Stock Exchange Corporation (TWSE). Daily price data of SSE 50 index constituent stocks were taken from the Shanghai Stock Exchange (SSE).

The remainder of this paper is organized as follows: In section 2, a brief introduction to some related works is provided. Section 3 presents the proposed STCNN-RN and GACG method, and the experimental results are provided in section 4. Finally, the conclusion of this paper and future research suggestions are provided in section 5.

II. RELATED WORKS

In this section, we explain our learning-to-reason model and how we perform anomaly detection through deep learning and explainable artificial intelligence approaches.

A. LEARNING TO REASON MODEL

We are convinced that existing knowledge and information exist in financial market data. In financial market information, various financial commodities have interactive relationships with the financial market, and various relationships operate between different financial commodities. The learning-to-reason model can reason about (and discover) the relationships between various objects or entities. This learning-to-reason model entails the process of discovering whether the system has meaningful patterns of information flow or data transformation [24].

To analyze the relationships between time series data and to model such data, some researchers have used traditional statistical methods and complex mathematical models. Podobnik *et al.* [25] proposed a method based on detrended cross-correlation analysis in physics, physiology, and finance to analyze the relationship between two sequences; their method may have added diagnostic capabilities to the statistical methods that were current at the time. Wang and other researchers [26] conducted research on the China stock market. From the perspective of statistical analysis, the authors used the detrended cross-correlation analysis between the return series of the China A-share and B-share markets and found that the system has long-term and short-term cross-correlations. Li and Liu [27] conducted research on cross-correlations between the agricultural commodity markets and the oil markets. The authors also used the DCCA method to discover that high oil prices caused the food crisis between 2006 and 2008. Kullmann *et al.* [28] sought a time correlation of returns between New York Stock Exchange stocks and studied whether the return of one stock can affect the return of another stock at different times.

The aforementioned research sought interactive relationships between various time series data, but those studies only modeled the cross-correlation between two sequences. Because the financial market is a complex and changeable environment, we require a model that can learn from multiple time series data sets to capture the complex interactions of financial market data. To surmount the previously discussed shortcomings, a novel model called RN [12] was proposed. RN is a neural network model proposed by DeepMind in 2017 to solve the relational reasoning problem. Most of the features of RNs focus on pattern recognition and classification, such as object detection, few-shot learning [29], image recognition [30].Just as CNNs have spatial translation invariance, RNs are inherently capable of relational reasoning. By constraining the functional form of a neural network, an RN has the core common properties of relational reasoning. Scholars have researched image recognition by combining an RN with a VQA system [31] The system performed exceptionally well with both three-dimensional-rendered objects and a text-based VQA data set.

In light of the aforementioned research, we observe that the learning-to-reason model has some deficiencies. Most of the current research on RNs has focused on image tasks for learning, but few studies have been published in the financial field. Much can be improved in the existing RN model. Because the tasks that the existing RN must address are pattern recognition and classification, most tested data sets consist of images, and thus, the relation module focuses on the spatial relationships among data. We contend that as long as different convolution operations are used on the input data, the relation module can be arbitrarily mobilized, meaning that both spatial and temporal relationships can be addressed. However, given the success of the RN in pattern recognition and classification and its ability to process spatiotemporal data, we propose that the RN has the ability to process financial data in multiple time series. Therefore, we further contend that the RN can be used to learn the characteristics of the inference problem between objects and to discover the interactive relationships between different time points relevant to companies.

B. ANOMALY DETECTION USING DEEP LEARNING

Anomaly detection technique can identify anomalies, novel data, and outlier data within vast quantities of data. Zenati *et al.* [32] presented high-performance generative adversarial networks (GANs) for anomaly detection using image data sets and network activity data sets. Leangarun *et al.* [33] demonstrated a model that merged a long short-term memory network (LSTM) and GANs to detect stock price manipulation. This was the first study that used LSTM-GANs to investigate stock price manipulation using time series data. However, Wang *et al.* [34]found that a company's characteristic features can effectively be used to distinguish between manipulated and nonmanipulated stocks. Published models have rarely incorporated this type of discriminatory feature. The authors applied an RNN-EL model for stock price manipulation problems using data sets with trading data and characteristic company features. Islam *et al.* [35] proposed detecting illegal insider trading of stocks by using proactive data mining on illegal insider trading cases and historical stock volume data. Furthermore, this research was the first such study of illegal insider trading using real cases.

Schreyer *et al.* [20] argued that the general anomaly detection method is to discover existing anomaly patterns from existing accounting data. Although a set of manual rules for catching anomalies can be learned from these existing anomaly patterns and the effect can be exceptional, it is to be expected that fraudsters will gradually discover methods to avoid these anomaly detection tools. Therefore, the authors proposed a method based on deep autoencoder neural networks to detect anomalous accounting data. Their results proved that the f1 scores obtained by this method were higher than those of the benchmark method and the false alarm rate was also lower than that of the benchmark method. Roy and other researchers [19] conducted research on anomaly detection in credit card data. They sought to discover anomalies in credit card fraud from existing credit card transaction data and historical customer data, thus providing a solution to the problem of credit card fraud detection. They proposed a deep neural network model to solve the problem of credit card fraud based on deep learning methods. It was found that the effect of LSTM and GRU is significantly better than that of a typical neural network for distinguishing abnormal transaction data from typical transaction data.

Few studies have been published on anomaly detection for time series data in the deep learning field; the present paper explains a deep learning method that can discover anomalous time points for stocks and companies in time series data. Few studies have applied deep learning methods to detect fraud in accounting data or assess whether credit card transaction data contain fraudulent data. To date, no studies have used deep learning to discover anomalies in stock data.

C. EXPLAINABLE ARTIFICIAL INTELLIGENCE APPROACHES

Deep learning is developing rapidly in the early twenty-first century. Deep learning tasks (e.g., natural language processing, image processing, recommendation systems, and anomaly detection) have been distinguishing themselves by high performance levels. Among typical deep learning tasks, image processing is the most commonly studied; it is a core research field of computer science and engineering. Of the successful applications of deep learning, training CNNs to recognize patterns and images has received considerable attention. CNNs that recognize patterns and images tend to encounter problems and vulnerabilities, such as some outliers or adversarial examples that can confuse neural networks.

Therefore, common algorithms and methods can be applied with neural networks [36]of interpretability and model inspection. Those methods can help us understand how a model works. Studies in a variety of fields have proven that intrinsically interpretable models can look at internal model parameters and make self-interpretations.A considerable number of studies of counterfactual explanations have changed some of the features to change predicted outcomes in a relevant manner. Several representative methods have been published for interpretable models. For example, Ribeiro and colleagues [37] proposed the LIME, in which this method interprets the prediction of any classifier in an interpretable and faithful manner by learning a local interpretable model based on the prediction. Through experiments, the authors proved that the effectiveness of interpretation can be demonstrated in various situations that require trust. In 2018, Ribeiro *et al.* [38] proposed an algorithm that can explain a black-box model with a high-probability guarantee and proved that the algorithm can predict the behavior of the model with less-than-usual effort and higher-than-usual accuracy. Guidotti *et al.* [39] considered decision-making systems with high accuracy but ambiguous judgment reasons. To explain these black-box subsystem decisions, the authors required some interpretation tools to reveal the reasons for predicting variables to make specific decisions. Therefore, LORE was proposed, which can provide interpretable and faithful explanations. The experiment results proved that the proposed method is superior to the benchmark method.

Some scholars have found that anomaly detection studies have not explained to investors why such anomalies are caught; the user cannot understand the reason for the formation of the anomaly. We expect that in addition to catching anomalies related to stock time series data, we can also explain the anomalies that were caught, enabling investors to understand the reasons for anomalies.

The preceding research summary indicates that the linear models of traditional finance do not completely fit the relevant theories. Deep learning models enable linear models to accurately simulate these theories, indicating that existing complex neural network models have a certain generalization ability. Most studies have employed deep learning methods to identify the interactive relationship between two sets of financial time-series data. Our contribution involves determining the interaction between various companies at multiple time points. Moreover, most research on RNs has focused on image tasks for learning, but few studies have investigated anomaly detection for time-series data in the financial field. Our proposed research framework is the first to combine RNs to model the complex cross-correlations between various companies in stock markets.

Financial market data do not have exact labels, making them dissimilar to text or image data when used in supervised learning. Therefore, traditional neural networks have some difficulty when training with such data. Our proposed STCNN-RN can learn the complex correlations between multiple financial time-series data sets; by using genetic algorithms with a constrained gene (GACG) to discover the time points for outlier companies, we can employ these outlier points to identify the abnormal situation.

Finally, most anomaly detection studies have been unable to explain to investors why such anomalies are identified and have provided investors with few crucial details; thus, the user cannot understand the reason for the formation of the anomaly. We used the local interpretable model-agnostic explanation (LIME) model to analyze and explain the causes of anomalies, thus helping investors to understand the market situation.

III. PROPOSED METHOD

In this part, we first introduce the overview of our system in Section 3.1, and we introduce the STCNN-RN and GACG, our main proposed methods in detail in Section 3.2 and Section 3.3, respectively.

A. SYSTEM OVERVIEW

Fig. 1 displays an overview of our system. The ultimate goal of the system is to identify the most abnormal time points in the financial time-series data. The anomaly information that is caught must include which companies have anomaly phenomena at which times. Moreover, these caught anomalies

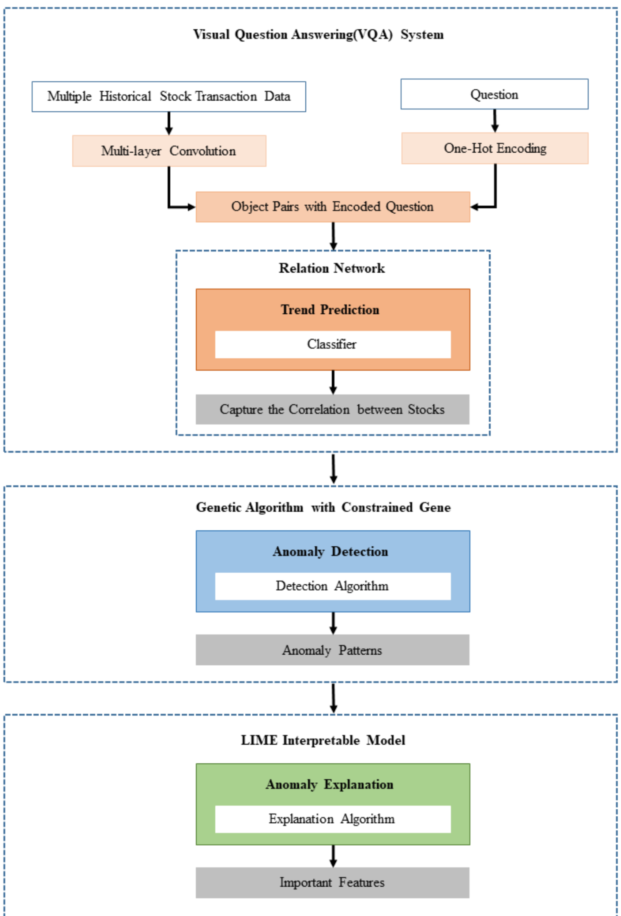


FIGURE 1. Overview of our system.

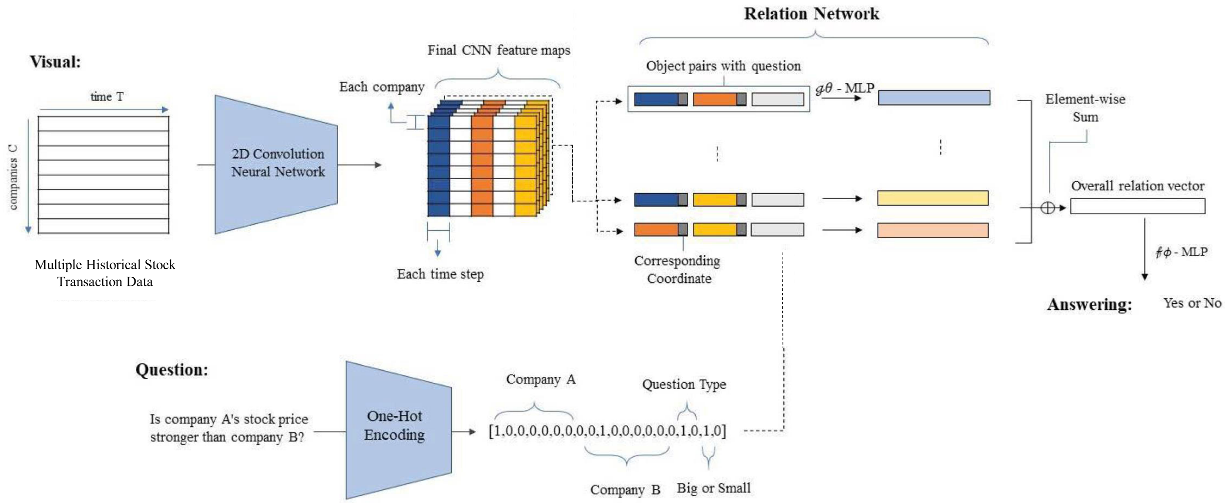


FIGURE 2. Network structure of the STCNN-RN.

should be explained reasonably and clearly, enabling system users or investors in the stock market to grasp the causes of the anomalies.

The main topic of this paper is anomaly detection. To perform anomaly detection on financial market data, two major problems must be addressed: (1) how to select anomaly time points from the total number of companies and times and (2) how to analyze the causes of the abnormalities in the selected time points and determine the types they belong to.

The three major parts of the system we designed can solve these two problems well. The first part is trend prediction (orange area in Fig. 1). For trend forecasting, we constructed a problem network structure that combines RNs and a visual question answering (VQA) system to capture the trends of diverse companies at various times and the interaction between them. We employed this approach because we believe that forecasting stock trends is more feasible than forecasting stock prices, and the learning task is more meaningful. After showing that our model can better fit the data, we expanded the number of stocks it learned from, expanding from individual stocks to the entire stock market. This can help investors to focus on the stocks they are interested in and subsequently gain insight into stock market trends.

The second part of the system involves anomaly detection (blue area in Fig. 1). In the preceding paragraph, we outlined how we fit the data to our model. After the data were fitted to our model, we introduced traditional genetic algorithms to improve and optimize the model. By optimizing and improving the model with traditional genetic algorithms, we were able to obtain genetic algorithms with constrained genes (GACGs) to meet the problem settings of our experiments. We used GACGs to conduct anomaly detection experiments on the VQA architecture combined with the input of the RN model. The GACGs identify the companies and the time points that are the strongest outliers in the model fit; thus, we endeavor to solve the first problem in our research through the first and second parts of our system.

The third part of the system involves anomaly explanation (green area in Fig. 1). We introduced the LIME model to explain the abnormal patterns identified in the first part and the second part. Moreover, using the introduced model to evaluate those features is more important for the model's judgment; we call these factors "key factors." By providing such information, (1) we hope that investors can more clearly understand the trend of the entire market and anomalous phenomena; (2) we address the second problem in our study.

B. STCNN-RN

In the previous chapter, we mentioned that the application of RNs combined with a VQA system for pattern recognition and classification tasks and relational reasoning problems can be successful because an RN considers the relations between objects in unstructured data. Our RN can be combined with a VQA system to match strengths and weaknesses. The combination can address volatility question types to fit financial market data to capture the interaction of an entire financial market. We combine the RN with a VQA system to meet our problem requirements and achieve the ultimate goal of our research. Our network architecture was inspired by the RN [12] proposed by DeepMind in 2017, which can process unstructured data such as an image or a series of sentences and implicitly infer the relations between the objects contained in it. We thus propose the STCNN-RN, which has the ability to process spatiotemporal data and manage the complex interactive relationships between the multiple time series.

As demonstrated in Figure 2, we constructed a problem network structure that combines RNs and a VQA system to capture the trends of different companies at various times and the interactions between them. This is because we believe that forecasting stock trends is more straightforward than forecasting stock prices and the learning task is more meaningful. After proving that our model delivered excellent data fitting, we expanded the number of stocks it learned from one stock to the entire stock market. This can help investors pay attention to the stocks they are interested in and gain insight into stock market trends.

A typical RN operates on objects in the simplest possible form, and thus it does not explicitly operate on images or natural language. The main contribution of the RN is to provide enough flexibility such that relatively unstructured inputs (such as CNN or LSTM embedding) can be regarded as a set of objects in an RN. Although an RN would typically take the object representation as input, the system has no requirement to specify the semantics of the object, and the RN learning process can induce upstream processing, thereby generating a set of useful objects from the distributed representation. To explain the operation mode and principle of our proposed model in detail, we divide our model into four parts for the explanation, namely Visual, Question, RN, and Answering.

1) VISUAL

The RN-with-VQA system has two inputs; a time series data set serves as one of the inputs of our model in the visual part, and this input also serves as one of the objects of our model. The CNN can convolve the returned data with the time series data size of $C \times T$, which C is the number of companies and T is the length of time. Although we use the two-dimensional (2D) convolutional layer, the order of each company in the input time series matrix does not matter. Therefore, it is illogical to perform convolution from top to bottom in the direction of the company axis. After 2D convolution, the information of each company can be collected according to the output unit setting of the previous convolution layer and then mapped to a one-dimensional feature map vector with different channel sizes. This may result in the loss of information about each company because it is a weighted sum of all companies. Consequently, we use a one-dimensional (1D) filter of size (1,w) for the 2D convolution operation, which is the window size of a 1D filter. The size of the company axis of the filter is locked to 1, and the 1D convolution operation is simulated by the length w of filter. We can obtain the feature map of input data through Eq. 1:

$$F_{k\times T'} = f\left(I_{C\times T} * k + b\right) \tag{1}$$

where $F_{k\times T'}$ represents the feature maps of the final convolutional layer, $I_{C\times T}$denotes the input data of the model, k represents the kernels (which can also be called filters) in the final convolutional layer, and b indicates the bias of the convolution layer. We know nothing about what specific image features should constitute the object. Therefore, after convolution of the image, the feature map is marked with arbitrary coordinates indicating its relative spatial position and is regarded as an object of the RN. This means that ''objects'' can include backgrounds, specific physical objects, textures, combinations of physical objects, and various other items. This provides the model with great flexibility in the learning process.

2) QUESTION

As we described previously discussed, this model has two inputs; one input falls on the visual part, and the other input falls on the question part. The text or sentence can be used as the input data of our question part. In this part, we illustrate the existing research in the context of bottlenecks and argue that the system has a relative and interactive relationship between different financial commodities in the time series data on the financial market. Therefore, we use two question types to learn and model the stock trends. These questions are based on the volatility of the stock and the strengths and weaknesses of the stock. The question with stock volatility is usually not much different from the following example: we can ask ''Is the volatility of company A larger than the volatility of company B?'' or ''Is the volatility of company A less than the volatility of company B?'' If the problem is related to the strengths and weaknesses of stocks, the question becomes ''Is Company A stronger than Company B?'' or ''Is Company A weaker than Company B?'' and so on. The system always has four methods to ask each question.

Whether the system has a relationship and meaning between objects depends on the question. To obtain the question encoded q,we use one-hot encoding to encode the question. The order of the encoding results is encoded according to the order of words. We can encode the two companies in the question, and then we can encode the types of question. We have two questions, one concerning the volatility of the stock and the other the strengths and weaknesses of stock. Finally, we encode the condition types to which the problem belongs. In one-hot encoding, the presence of the number ''1'' indicates that the feature exists, and the number ''0'' indicates that the feature does not exist.

3) RN

The system has two inputs in the module of the RN, which are the feature maps from the CNN and the problem encoding results from one-hot encoding. Now we must build an RN module. The input to this network is a set of ''objects'' $O = \{o_1, o_2, \ldots, o_n\}$. For example, the number of objects depends on the sequence length of the final CNN feature map. After convolution, each feature map must be 2D (which retains the company axis and time axis), the target time step and company information become a vector (which consists of the three-dimensional collected values output along with the 2D convolution layer). This retains company information by convolving each company separately. In addition, because the order of companies in the matrix is arbitrary, we avoid convolution in the direction of the company axis at the same time. If we want to extract a set of objects for relation calculation, we must use the third axis value of the final CNN as the object. We must mark these objects with time and company coordinates, express them in the form of a key vector, and feed the object pairs to the RN module. Then we use two MLPs to infer the object relationship. First, g_θ - MLP can calculate each object-to-object relationship and represent part of the relationship. The f_ϕ - MLP can sum element-by-element from the output of the g_θ - MLP and calculate the overall relationship. Finally, the RN module can learn the relationships between all the object pairs, embed the question (considered as a condition), and integrate all these components to answer the question. The simplest function of the RN model is formalized as

$$RN(O) = f_\phi\left(\sum_{i,j} g_\theta\left(\left(o_i, c_i, o_j, c_j\right), q\right)\right) \tag{2}$$

where the input to this network is a set of "objects" $O = \{o_1, o_2, \ldots, o_n\}$, o_i, o_j represents each object-object pair, c_i, c_j denotes the corresponding coordinate of each object, q is the question encoding,g_θ stands for the MLP with parameters θ to calculate partial relations, and f_ϕ indicates the second MLP with parameters ϕ to calculate the entire relation. The RN module can consider all pairs of objects, the embedding of the question, the time tags, and the company coordinate of objects; the RN can integrate all these components to answer the corresponding questions.

4) ANSWERING

In the answering part, we can obtain the output after g_θ - MLP, which is called the partial relation. The sums of all partial relation element can be written as a matrix, and that matrix is named the overall relation vector. This overall relation vector V_{or} can enter a multilayer feed-forward neural networkf_ϕ, and then the last layer of the feed-forward neural network must be processed through the sigmoid activation function as Eq. 3 to limit the output value range between 0 and 1:

$$softmax(t) = \frac{1}{1 + e^{-t}} \tag{3}$$

where t denotes the output of f_ϕ. The output of the model can also be predicted for the corresponding question to obtain the prediction answer $\hat{A}$ as in Equation 4.

$$\hat{A} = softmax(f_\phi(V_{or})) \tag{4}$$

To measure the accuracy of our proposed model prediction, we can use binary cross-entropy to calculate the gap between the predicted answer and the actual answer. The formula of binary cross-entropy is

$$Loss_{BCE} = -\frac{1}{N}\sum_{i=1}^{N} A_i * \log(\hat{A}_i) + (1 - A_i) * \log(1 - \hat{A}_i) \tag{5}$$

where N represents the output size of $\hat{A}$, which indicates the size of each batch. $\hat{A}$ denotes the predict answer and A regarded as the actual answer. When we are training the model, we will use Equation 5 to minimize the loss of our model (which is equivalent to optimizing learning).

The preceding passages fully explain the operation and principles of our RN. Because of the successes of previous RNs and the ability of RNs to process spatiotemporal data, we introduce the RN into our research. We are convinced that some interactions exist between various financial commodities, and thus, we use the RN to learn stock trends.

Algorithm 1 Genetic Algorithm With Constrained Gene for Anomaly Detection

Input:
$A_{C\times C\times T}$: The accuracy matrix generated from the proposed model, which is consist of the accuracy result of all C companies during a period of time T;
N: Population size for each generation;
C_{rate}: Cross rate, which is mating probability for chromosome crossover;
M_{rate}: Mutation rate, which is the mutation probability of each chromosome;
Output:
Chromosome $p_{highest}$: The best chromosome in all genarations.

Generate initial population $\mathrm{P}_0 = \{\mathrm{p}_0, \mathrm{p}_1, \ldots, \mathrm{p}_{N-1}\}$
Evaluate the fitness values of population P_0 using $A_{C\times C\times T}$ by Equation 9
Find the highest fitness values of *Chromosome* $p_{highest}$ by Equation 10
Calcuate the accuracy $GA_{p\_highest\_acc}$ of Chromosome $p_{highest}$ by Equation 11
Find the *Rule-based*$_{acc}$ by Equation 12
while ($GA_{p\_highest\_acc}$ < *Rule-based*$_{acc}$) **do**
  $\mathrm{P}_{new} = []$
  Append (Select the top$k$fitness values of chromosomes p_{top_k}) into P_{new}
  Select N-k chromosomes from population P according to the fitness values
  for each chromosome $i \in N$-k chromosomes **do**
    Crossover chromosome i with C_{rate}
    Mutation chromosome i with M_{rate}
    Check up and Limit the Gene i
    Append (chromosome i) into P_{new}
  end for
  $\mathrm{P} = \mathrm{P}_{new}$
  Evaluate the fitness values of population P
  Find the highest fitness values of chromosome $p_{Highest}$
  Calcuate the accuracy $GA_{p\_highest\_acc}$ of *Chromosome* $p_{highest}$
end while
return *Chromosome* $p_{highest}$

After appropriately training the proposed model, we can use the proposed model to capture the interactions of financial commodities in the entire financial market. To retain company and time information, we have added some features to the RN. For example, our RN performs a 1D convolution operation on a 2D convolution layer and adds corresponding coordinates to the feature maps of the final convolution layer.

C. GACG

The preceding subsection completely describes the characteristics of the RN, its operational methods, and the principles of the model. Through the complete training of the RN to fit the data of the entire financial market, the RN can learn the interactive relationships between financial commodities to ensure that the model can fully grasp the relationships in the entire financial market. For any set of data, after we have trained this RN, we hope to discover the data that cannot fit our model. Therefore, we hope to discover training data that cannot be learned by our model in the anomaly detection phase. A particular subset of captured training data contains information about a particular company at a particular point in time. The ultimate goal is to discover the most extreme outliers among the training data. We contend that the data that cannot be fitted by the model may describe a certain period in the financial market that is markedly different from most periods for financial products in the entire market. At the same time, these captured data are different from the usual changes, meaning that our RN cannot simulate these data. However, how to discover the most extreme outlier data and the most anomalous training data among multisource financial time series data is a complicated optimization problem. Therefore, we introduce a genetic algorithm to solve this problem. We can make changes and optimizations based on the GACG to meet our problem setting and achieve the optimal effect. To clearly explain the operation of the algorithm and its principle, we can divide the algorithm we propose into six parts and describe and explain the detailed operation of each part. The six parts are generate, fitness function, selection, crossover, mutation and assessment, and limit the gene.

Before this genetic algorithm is executed, we must train the STCNN-RN so that it can achieve high training accuracy. To successfully execute the GACG, we must perform some preliminary operations and prepare the input data required by it. We must send the training data (segmented according to the date) to a properly trained STCNN-RN for evaluation. Through this method, an accuracy matrix $A_{C\times C\times T}$ is created, where C represents the number of companies, and T represents all dates of the training data. This accuracy matrix $A_{C\times C\times T}$ serves as the population in the actual real world solution space. As shown in Figure 3, we proposed genetic algorithm with constrained gene(GACG) to conduct anomaly detection experiments on the VQA architecture combined with the input of the relational network model. The genetic algorithm with constrained gene will find the companies and the time points that are the most outlier in the model fit, so we hope to solve the first problem in our research through the first part and the second part.

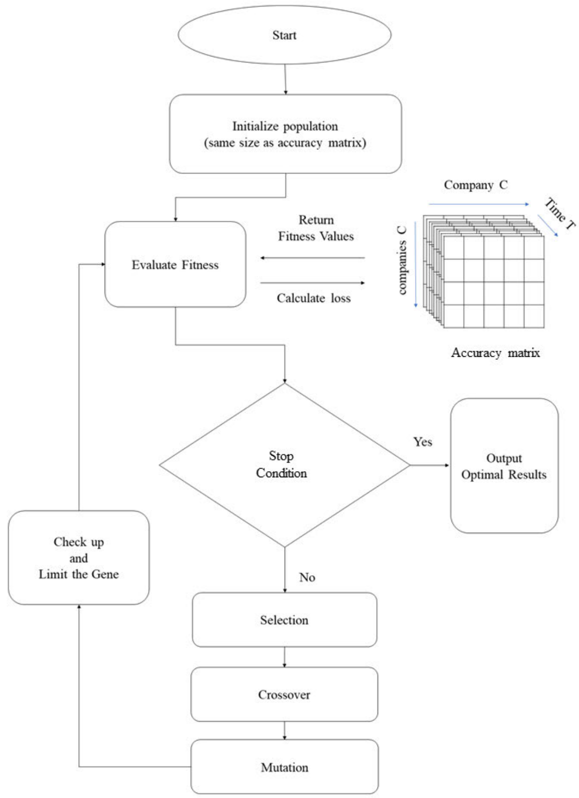


FIGURE 3. Flow Chart of the GACG.

1) GENERATE

After we have completed the pre-operation of the genetic algorithm, then we officially start the first step of the genetic algorithm. The first step of the genetic algorithm is to generate population. Firstly, we need to decide on the representation we use to represent the solution of genetic algorithm. The incorrect representation will cause the genetic algorithm to perform poorly. The most simple and frequently used representation in the past genetic algorithms has been the binary representation. This form of representation is relatively easy to use in the computing space to express solutions in a way that the computing system understands and operates. In this study, we use a binary representation of $C \times C \times T$ elements to represent a solution. In this binary representation of chromosome p, the $p_{i,t}$ th element represents the training accuracy rate of the i th company at the time point t. We will define the $p_{i,j}^t$ th element as follows:

$$p_{i,j}^t = \begin{cases} 1, & anomaly \\ 0, & normal \end{cases} \tag{6}$$

If the $p_{i,j}^t$ th element is an anomaly, the definition returns (1), and otherwise, the definition returns (0). In our research, the population is defined as a subset of problem solutions. In the process of generating a population, we can ensure the diversity of the population to avoid the problem of premature convergence. We must repeatedly experiment to determine the optimal population size because any excessively large population can cause GACG's operation speed to be unacceptably slow. Conversely, any excessively small population can cause the problem of insufficient population diversity. After deciding the size of our population, we only fill a few reasonable solutions in the set of all initialized solutions and we fill the rest of the population with random solutions.

2) FITNESS FUNCTION

After generating an initial population according to the population size, we must evaluate the quality of each solution. Therefore, we introduce the fitness function we designed to calculate the fitness values of the population to evaluate the gap between the population and the target solution. The fitness function here mentioned is a function that takes candidate solutions to the problem as input and produces as output—how acceptable the "fitness" is of the solution to the problem that is under consideration. The calculation of the fitness value is repeated in genetic algorithms and must be adequately fast. Slow calculation of the fitness values may adversely affect genetic algorithms by making them extremely slow. However, the fitness function must have the following characteristics: the fitness function should be calculated fast enough, and it must quantitatively measure the suitability of a given solution or how to generate individuals from the given solution. In some cases, due to the inherent complexity of the problem at hand, it may not be possible to directly calculate the fitness function. In this case, we can perform fitness approximation to meet our needs. Before entering the fitness function we designed, we must decode each solution.When the value of $p^t_{i,j}$ is 1, we choose to set the accuracy to 0; otherwise, if the value of $p^t_{i,j}$ is 0, we will get the accuracy rate from the accuracy matrix $A^t_{i,j}$ corresponding to the same position of $p^t_{i,j}$.

$$Gene_{values(p^t_{i,j})} = \begin{cases} 0, & \text{if } p_{i,t} = 1 \text{ or } p_{j,t} = 1 \\ A^t_{i,j}, & \text{otherwise} \end{cases} \tag{7}$$

Then, we sum all the accuracy values of $p^t_{i,j}$. $p_{values(p_x)}$, which can be expressed as the following equation:

$$p_{values(p_x)} = \frac{\sum_{t=1}^{T}\sum_{i=1}^{C}\sum_{j=1}^{C} Gene_{values(p^t_{i,j})}}{C*C*T-(C*C*T*0.05)} \tag{8}$$

where p_x represents a chromosome of the population, C denotes the number of companies, T stands for all the transaction day of the training data and $p^t_{i,t}$ describes the training accuracy of the i th company and j th company at the time point t.After obtaining all the accuracy rates, to obtain the average of all the accuracy rates, we can obtain all the accuracy rates divided by the number of $C*C*T-(C*C*T*0.05)$. To evaluate the fitness values between solution and target, we have designed a fitness function that meets our problem setting. The fitness function we designed is as follows:

$$f(p_{values}) = p_{values} + (1e^{-3}) - min(p_{values}) \tag{9}$$

Equation 9 makes the calculated fitness values $f(p_{values})$ large when the accuracy values of chromosome $p_{values(p_x)}$ are large. To allow every chromosome in the population to have a chance of producing offspring, we add a small number to the fitness function, and consequently, the smaller p_{values} becomes a smaller number instead of 0.

3) STOP CONDITION

After calculating the fitness values belonging to each chromosome, we must evaluate whether any chromosome exists that meets the suspension conditions in this generation. Before evaluating whether any chromosome meets the suspension conditions, we must first discover the chromosome with the highest fitness values in this generation, which then serves as a candidate for later use to measure whether the stop conditions are met. The equation for discovering the chromosome of highest fitness values $p_{highest}$ is expressed as follows:

$$p_{highest} = max(f(p_{values})) \tag{10}$$

After finding the chromosome with highest fitness values $p_{highest}$, we can introduce it into the equation $GA_{p\_highest\_acc}$ to calculate its accuracy and set the accuracy as $GA_{p\_highest\_acc}$:

$$GA_{p\_highest\_acc} = p_{values(p_{highest})} \tag{11}$$

In addition, we must discover a benchmark that satisfies the stop conditions. We discover the benchmark for the stop condition by discovering the worst accuracy rate of 5% in the accuracy matrix $A_{c,t}$ Moreover, we set these accuracy rates to 0, sum all the accuracy rates, and then average them. The previously discussed steps are how we look for Greedy accuracy *Greedy*. It can be expressed as the following equation:

$$Greedy = p_{values(p_{greedy})} \tag{12}$$

After finding the candidate chromosome $GA_{p\_highest\_acc}$ of the stop condition and the baseline chromosome *greedy*, we can compare the differences between them. If the accuracy of the candidate chromosome $GA_{p\_highest\_acc}$ exceeds the accuracy of the baseline chromosome *greedy*, we stop the operation of the GACG. Conversely, if the candidate chromosome $GA_{p\_highest\_acc}$ is less than the baseline accuracy rate *greedy*,we resume executing the GACG.

4) SELECTION

When the candidate chromosome $GA_{p\_highest\_acc}$ is less than the baseline accuracy rate *Greedy*, we can enter the selection part. In this part, we can select the top $k$fitness values of a chromosome to retain for the next generation. This allows them have the opportunity to reproduce the chromosome, and thus, our candidate chromosome can obtain a higher accuracy rate. Then, we can use fitness values to calculate the probability of the screening chromosome and use this probability to select N-k chromosomes.The equation for calculating the probability of screening chromosome is as follows:

$$P(f(p_{values(p_i)})) = \frac{f(p_{values(p_i)})}{\sum_{i=1}^{N} f(p_{values(p_i)})} \tag{13}$$

where $=$ p_i represents a chromosome of the population, $f(p_{values(p_i)})$ stands for the fitness values of the chromosome, and Ndenotes the size of the population for each generation. Then we can screen out N-$k$chromosomes equation according to the calculated probability $P(f(p_{values(p_i)}))$. Each of these N-k chromosomes must enter the crossover part.

5) CROSSOVER

First, we can make each of the *N-k*chromosome take turns as the parent, and we call it "father." We can generate a random number and compare it with cross rateC_{rate}. If the random number is larger than the cross rate C_{rate}, then it cannot enter the stage of cross over and the father's chromosomes directly become the chromosomes of the next generation. Conversely, if this random number is less than the cross rate C_{rate}, we can officially enter the stage of cross over. After entering the crossover part, we can randomly select one of the *N-k* chromosomes as the mating target and we call it "mother." Further, we can discover and remember the mother gene position of 1. Suppose we set the number of anomalous time points to 60. Then, we can extract the half of gene position 1 among the mother's gene, and half the gene positions are extracted from the father's gene. For the remaining gene positions, we fill in 0 as the gene. We can combine the extracted mother and father genes to form new genes. This describes the mating process of chromosomes.

6) MUTATION

We take the chromosome that just emerged from the cross over and enter it into the gene mutation part. First, we generate a random number, and we can determine whether we must mutate genes $M_{rate}$for each gene. One can judge whether a mutation is required by using a randomly generated number and comparing it with the mutation rate M_{rate}we set. If this random number is greater than the mutation rate M_{rate}, we do not enter the mutation procedure, and the chromosome can be directly used as the new chromosome of the next generation. Conversely, if this random number is less than the mutation rate M_{rate}, we must enter the stage of gene mutation. In the stage of gene mutation, we can turn the original 1 gene into 0. Conversely, genes that were originally 0 are set to 1.

7) ASSESS AND LIMIT THE GENE

In this part, we ensure that in each chromosome, the number of gene value 1 matches the number of anomalous time points we set. We have specially set up the system to assess and limit the gene. After each chromosome has gone through selection, mating, and mutation, the system can confirm that the number of gene value 1 matches the number of anomalous time points we set. First, we can assess in each chromosome whether the number of gene value 1 matches the number of anomalous time points we set. If the number of gene value 1 matches the number of anomalous time points, the chromosome can be regarded as the new chromosome of the next generation. Conversely, if the number of gene value 1 does not meet the number we set, we limit the number of genes. First, we discover and record the positions of all 1 in the gene and then extract the number we set from them as the new ones; then, we set the remaining positions to 0. After this stage is completed, the new chromosome becomes the population P_{new} of next generation.

When we obtain a new population P_{new},we can recalculate fitness values of the new population. Then, we discover the candidate chromosome $GA_{p\_highest\_acc}$ of the new population P_{new} to compare with the baseline accuracy rate *Greedy* to meet the suspension conditions until the stop conditions are met. In the end, the algorithm can return an optimal solution. The best solution is to achieve the best accuracy after deleting the anomalous time point.

IV. EXPERIMENTS

To ensure and verify the validity of our proposed model, we conducted three experiments to evaluate and measure the proposed model. Before discussing the experiments, we describe the data set used in the experiments. The baseline methods used to compare the different models are defined. After a detailed description of the experimental setup, the experimental results of the experimental setup are displayed and the three experiment's results are discussed.

A. DATA SET DESCRIPTION

We used four data sets in three experiments: the S&P 500 constituent stocks Top 20 Tech Companies data set, S&P 100 constituent stocks data set, FTSE TWSE Taiwan 50 index constituent stocks data set, and SSE 50 index constituent stocks data set.

S&P 500 constituent stocks Top 20 Tech Companies data set: As shown in Tables 1, we collected the minute data of S&P 500 constituent stocks from 2015 to 2016. S&P 500 constituent stocks contain five features, which are open, high, low, close, and volume. We can select the company in the S&P 500 constituent stocks according to its published weight. Therefore, we selected the 20 technology companies in the S&P500 with the greatest weights as our first data set. The number of stocks in the 2 years is different. After discarding some incomplete stock data, the number of stocks in 2015 and 2016 are 483 and 490, respectively. In the process of screening stocks, we selected 20 technology companies that coexisted during the 2 relevant years.

TABLE 1. The company list of S&P 500 constituent stocks's top 20 tech companies dataset.

Ticker Symbol	AAPL, MSFT, V, MA, INTC, CSCO, ADBE, CRM, NVDA, ACN, AVGO, ORCL, IBM, TXN, QCOM, FIS, ADP, INTU, FISV, MU

The S&P 100 constituent stocks data set: As shown in Tables 2. In Experiment 1, the second data set we used is the minute stock price data of S&P100 constituent stocks. Minute price data of S&P 100 constituent stocks were taken from Wharton Research Data Services Trade and Quot-e (WRDS TQA) [23]. To compare the effects between the S&P 500 constituent stocks Top 20 Tech Companies data set and S&P 100 constituent stocks data set, we took the years shared by the two data sets and finally screened out the 2015 and 2016 data for model training. Therefore, when screening

TABLE 2. The company list of S&P 100 constituent stocks dataset.

Ticker Symbol	AAPL, ABBV, ABT, ACN, AIG, ALL, AMGN, AMZN, AXP, BA, BAC, BIIB, BK, BMY, BRKB, C, CAT, CELG, CL, CMCSA, COF, COP, COST, CSCO, CVS, CVX, DIS, EMR, EXC, F, FB, FDX, FOXA, GD, GE, GILD, GM, GOOG, GOOGL, GS, HAL, HD, HON, IBM, INTC, JNJ, JPM, KMI, KO, LLY, LMT, LOW, MA, MCD, MDLZ, MDT, MET, MMM, MO, MON, MRK, MS, MSFT, NKE, ORCL, OXY, PEP, PFE, PG, PM, QCOM, RTN, SBUX, SLB, SO, SPG, T, TGT, TWX, TXN, UNH, UNP, UPS, USB, UTX, V, VZ, WBA, WFC, WMT

S&P100 constituent stocks, we also only took stocks shared in the 2 years. The list of S&P100 companies in the 2 years is different; we took out the stock price data of 90 companies that also existed in 2 years from the 100 companies.

The FTSE TWSE Taiwan 50 Index constituent stocks data set: The FTSE TWSE Taiwan 50 index is an index jointly compiled by the Taiwan Stock Exchange(TWSE) and the FTSE Index. The FTSE TWSE Taiwan 50 Index Components cover the top 50 listed companies in the Taiwan stock market by market capitalization, and it is highly correlated with the broader market. The minute price data of stocks were taken from the TWSE, and we collected data on intraday stock transactions from 2016 to 2017, including information such as opening, high, low, closing, and volume. Because the list of FTSE TWSE Taiwan 50 Index constituent stocks in 2016 and 2017 are different, we only took stocks shared in 2016 and 2017 as our training data. We screened 48 listed companies from 50 listed companies in Experiment 2.

The SSE 50 Index constituent stocks data set: Daily price data of SSE 50 index constituent stocks were taken from the Shanghai Stock Exchange (SSE). The SSE 50 Index is an index compiled by the SSE. It is regarded as an index representing the overall situation of the most influential companies on the SSE. It selects the most representative stocks with a large scale and excellent liquidity in the Shanghai stock market to form the index. The goal of compiling this index is to establish a large-scale investment index with active transactions that can be used as the basis for derivative financial instruments. The method of selecting component stocks is to comprehensively rank stocks based on market capitalization and turnover and to select the top 50 stocks to form a sample, excluding stocks that have abnormal market performance and are deemed inappropriate by the expert committee. Due to changes in the SSE's selection of constituent stock rules, the list of SSE constituent stocks has changed greatly. We collected daily trading data of constituent stocks in 2019 and 2020. We selected from 50 listed companies coexisting on the constituent stock list in 2019 and 2020 and finally selected 37 for our training and testing data sets in Experiment 2.

B. PERFORMANCE EVALUATION OF THE PROPOSED ANOMALY DETECTOR

An experiment was conducted to compare the performance of the STCNN-RN with other baseline models.

A multilayer deep neural network (multilayer DNN) is a model composed of multiple layers of density. We introduced a model composed of multiple layers of density as our experimental baseline model. The input of the visual part in this model is $I_{C\times T}$, but we chose to directly import the flattening layer to process this input. The input processing method of the question part is the same as that of the STCNN-RN. Then, we import the concatenated layer to process the output from the visual part and question part and connect four layers of density behind this concatenated layer. The output of the last dense layer passes through the activation function "sigmoid" before the answer A is returned. The process previously discussed is how this model works.

Another baseline model used in our experiment was the temporal convolution neural network–based relational network (TCNNRN). We applied financial time series data in a model based on 1D convolution. The difference between the TCNNRN and STCNN-RN is that in the visual part, the TCNNRN can use a multilayer 1D-CNN to process the input matrix to generate a set of objects for the RN module. After returning the data for convolution operation, we can connect the object pairs extracted from the final CNN feature map with the corresponding time coordinates and question one-hot encoding vector. Then, we feed the connected vector to the RN module to calculate the relation, and the final output is the corresponding answer A.

To verify the effectiveness of the GACG, we also proposed other baseline methods for determining genes, which are None, Random Choice, and Greedy. None is the method to measure the ability of our model to capture the relationship between different time series data, for which we introduced binary cross-entropy to calculate the performance of the model on the binary classification problems. To evaluate the ability of the GACG to catch anomalies, we introduced the random choice method. The concept of random choice is to randomly generate N chromosomes and then calculate the accuracy for each chromosome, after which we sum the N calculated accuracy and divide by N to obtain the accuracy of random choice. We also select the worst accuracy rate of 5% in the accuracy matrix and set the values to 0. Then we add the entire accuracy matrix $A_{C\times C\times T}$ and divide by $C \times C \times T$ to obtain the Greedy accuracy.

In Experiment 1, we used two data sets to train our proposed model. The two data sets are the S&P 500 constituent stocks Top 20 Tech Companies data set and S&P 100 constituent stocks data set. The purpose of using these two data sets was to verify the effectiveness of our proposed model. Moreover, we also want to prove that our model can learn the interactions between stocks regardless of having only a small amount of time series data or a large amount of time series data.

TABLE 3. Anomaly detection accuracy of existing models with S&P 100 constituent stocks data set.

Model	Anomaly Detection	Rolling Test 1	2	3	4	5	6	7	8	9	10	Average
Multi-layer DNN	None	0.5576	0.5067	0.7545	0.6108	0.6654	0.6054	0.6183	0.6355	0.6662	0.5168	0.6083
	Random Choice	0.5766	0.5251	0.7744	0.6299	0.6848	0.623	0.6353	0.6532	0.6847	0.5381	0.6415
	Greedy	0.5773	0.5258	0.7752	0.6305	0.6856	0.6238	0.6363	0.6542	0.6855	0.5388	0.6423
	Proposed GACG	0.5776	0.526	0.7757	0.6307	0.6858	0.6241	0.6367	0.6546	0.6859	0.5391	**0.6426**
TCNN-RN	None	0.5747	0.5103	0.5649	0.544	0.5138	0.5164	0.5508	0.5203	0.5219	0.5172	0.5334
	Random Choice	0.5812	0.516	0.5713	0.5501	0.5196	0.5222	0.557	0.5262	0.5278	0.523	0.5394
	Greedy	0.5819	0.5168	0.572	0.5508	0.5203	0.5228	0.5577	0.5269	0.5284	0.5237	0.5401
	Proposed GACG	0.8661	0.5192	0.5759	0.554	0.5228	0.5252	0.5612	0.5292	0.5305	0.5252	**0.5709**
STCNN-RN	None	0.7964	0.7664	0.7867	0.7799	0.795	0.7772	0.7698	0.7745	0.818	0.8127	0.7876
	Random Choice	0.8054	0.775	0.7956	0.7887	0.8039	0.7859	0.7784	0.7833	0.8272	0.8218	0.7965
	Greedy	0.8067	0.7761	0.7964	0.7896	0.8049	0.787	0.7796	0.7843	0.8282	0.823	0.7976
	Proposed GACG	0.8212	0.7907	0.8093	0.8027	0.818	0.8006	0.7928	0.7978	0.8429	0.8368	**0.8113**

TABLE 4. Anomaly detection accuracy of existing models with FTSE TWSE Taiwan 50 Index constituent stocks data set.

Model	Anomaly Detection	Rolling Test 1	2	3	4	5	6	7	8	9	10	Average
Multi-layer DNN	None	0.7721	0.829	0.741	0.792	0.7511	0.6959	0.7322	0.7179	0.7027	0.7167	0.7451
	Random Choice	0.7886	0.8467	0.7569	0.8088	0.7671	0.7107	0.7478	0.7331	0.7178	0.7321	0.761
	Greedy	0.7898	0.8479	0.7582	0.8104	0.7685	0.7122	0.7491	0.7345	0.7191	0.7338	0.7624
	Proposed GACG	0.8119	0.8715	0.7804	0.8323	0.7896	0.7319	0.7697	0.7547	0.7402	0.7564	**0.7839**
TCNN-RN	None	0.7331	0.7453	0.7171	0.7508	0.7233	0.7249	0.7521	0.715	0.7309	0.7111	0.7304
	Random Choice	0.7487	0.7612	0.7323	0.7668	0.7387	0.7403	0.7681	0.7302	0.7465	0.7263	0.7459
	Greedy	0.75	0.7626	0.7337	0.7682	0.7401	0.7418	0.7694	0.7316	0.7481	0.7278	0.7473
	Proposed GACG	0.7713	0.7851	0.7544	0.7898	0.7625	0.7624	0.7914	0.7516	0.77	0.7487	**0.7687**
STCNN-RN	None	0.852	0.8851	0.8599	0.8271	0.7417	0.8477	0.8803	0.851	0.9027	0.8797	0.8527
	Random Choice	0.8702	0.904	0.8782	0.8447	0.7576	0.8657	0.8991	0.8691	0.9219	0.8984	0.8709
	Greedy	0.8711	0.9046	0.8789	0.846	0.7592	0.8665	0.8996	0.8701	0.9224	0.8989	0.8717
	Proposed GACG	0.8941	0.9273	0.9008	0.8687	0.7809	0.889	0.9222	0.8935	0.9457	0.9212	**0.8943**

In Experiment 1, we used two data sets, which are the S&P 500 constituent stocks Top 20 Tech Companies data set and S&P 100 constituent stocks data set. We used the closing price to calculate the return data required by the model and then processed them into the input required by the model. The results in Tables 4 and Table 3 indicate that the extent to which the model can learn financial time series data affects the model's fitting accuracy. Based on the results of the experiments, we conclude the STCNN-RN has excellent fitting accuracy for the S&P 500 constituent stocks Top 20 Tech Companies data set and the S&P 100 constituent stocks data set. This means that the STCNN-RN has the ability to learn the financial time series data and is suitable for processing spatiotemporal data. For comparing the results of these models, we used bold fonts to highlight the highest anomaly detection accuracy.

To conduct the anomaly detection accuracy experiments detailed in Table 4 and Table 3, we used the random choice method and Greedy method, respectively, to screen our best solutions. The solution obtained by the STCNN-RN combined with the GACG achieved the highest accuracy after deleting all anomalous time points with companies, followed by the performance of the multilayer DNN combined with the GACG and the STCNN-RN combined with the GACG. Table 3 shows the STCNN-RN combined with the GACG has a higher accuracy rate than the other baseline models, and the anomaly accuracy rate ranges from 0.7907 to 0.8429 in most of the rolling tests. Table 4 has the same results, the anomaly accuracy rate ranges from 0.7809 to 0.9457. Therefore, we can draw the following conclusions from the results of Experiment 1: the STCNN-RN can accurately capture the interactions between different companies, and the STCNN-RN has the ability to process spatiotemporal data. Compared with other baseline models, the experiment with the STCNN-RN combined with the GACG proves that the model can discover the best solution to obtain a higher anomaly detection accuracy rate.

C. ANOMALY DETECTION IN VARIOUS FINANCIAL MARKETS

To prove that our model has a certain generalization ability, in addition to using the S&P 100 data set to verify the performance of our proposed model on the US stock market, we introduced FTSE TWSE Taiwan 50 Index Components and SSE 50 Index Components; they represent the Taiwan stock market and the Shanghai stock market respectively.

In Table 5, we list the outcome of subjecting three data sets to our proposed model and baseline models. In this experiment, the three data sets represented different stock markets. The component stock data set of the S&P100 Index represented the US stock market, the component stock data set of FTSE TWSE Taiwan 50 Index represented the Taiwan stock market, and the SSE 50 Index represented the Shanghai stock market.

It turns out that the proposed model can fit the stock markets of different stock markets well. Even when facing data sets with different data frequencies, the STCNN-RN fitting data accuracy was still better than that of the TCNNRN

TABLE 5. Anomaly detection accuracy of existing models with SSE 50 Index constituent stocks data set.

Model	Anomaly Detection	Rolling Test										Average
		1	2	3	4	5	6	7	8	9	10	
Multi-layer DNN	None	0.8352	0.793	0.7967	0.7815	0.8148	0.8447	0.8143	0.8352	0.8246	0.8092	0.8149
	Random Choice	0.9171	0.876	0.8817	0.8592	0.9167	0.9542	0.9043	0.9171	0.903	0.8807	0.901
	Greedy	0.9213	0.8815	0.8868	0.8642	0.9213	0.9569	0.9087	0.9222	0.908	0.8877	0.9059
	Proposed GACG	0.9527	0.9115	0.9174	0.8933	0.9536	0.9607	0.94	0.9542	0.9383	0.9175	**0.9339**
TCNN-RN	None	0.545	0.5355	0.5664	0.5119	0.5198	0.5268	0.5699	0.5843	0.5595	0.5741	0.5493
	Random Choice	0.728	0.6569	0.7353	0.6407	0.6581	0.6449	0.724	0.7537	0.7224	0.6588	0.6923
	Greedy	0.7284	0.666	0.7366	0.6492	0.6657	0.6467	0.7229	0.763	0.7173	0.6667	0.6963
	Proposed GACG	0.77	0.6918	0.775	0.6756	0.6939	0.6781	0.7606	0.7938	0.7653	0.6915	**0.7296**
STCNN-RN	None	0.8133	0.8448	0.938	0.8567	0.9092	0.8904	0.9077	0.9106	0.9035	0.9388	0.8913
	Random Choice	0.9446	0.9294	0.9716	0.9476	0.9603	0.9628	0.9626	0.9792	0.9496	0.9767	0.9584
	Greedy	0.9487	0.9364	0.9741	0.9515	0.9634	0.967	0.9663	0.9806	0.9566	0.9787	0.9623
	Proposed GACG	0.9822	0.97	0.9754	0.9857	0.9939	0.9985	0.998	0.9816	0.9959	0.9799	**0.9861**

or multilayer DNN. Moreover, this proved that the proposed model can learn the interactive relationships in different time series data regardless of whether it is intraday transaction data or daily transaction data and that it is more than capable of predicting future trends. However, the performance of the TCNNRN was slightly inferior to that of the multilayer DNN, possibly due to the use of the 1D-CNN to process input data to make object pairs, which prevents the model from learning how to predict stock trends.

Table 5 lists the accuracy results of the three models combined with different baseline methods of the best solution. Comprehensive assessment of the experiment results indicates that the STCNN-RN combined with the GACG has a higher accuracy rate than the other baseline models combined with different baseline methods and the anomaly accuracy rate ranges from 0.97 to 0.9985 in most of the rolling tests. The second highest performance was that of the multilayer DNN combined with the GACG, followed by the performance of the TCNNRN combined with the GACG. In this experiment, the time intervals in the three data sets were different, but their rolling windows and shift month methods were the same.Therefore, we once again proved the quality of the performance of our model in the learning tasks and complex correlation of multiple financial time series data in this experiment. This indicates that when facing various stock markets, data frequencies, or data intervals, the STCNN-RN combined with the GACG method has outstanding performance.

D. COMPARE DIFFERENT ALTERNATIVE MODELS

Tables 3-5 indicate that the solution obtained by the STCNN-RN combined with the GACG achieved the highest accuracy. Therefore, we used the STCNN-RN combined with the GACG method to perform comparisons with other benchmark models. We selected the one-class SVM (OC-SVM) [40] and self-organizing maps (SOMs) [41] as the control group.

The OC-SVM is an unsupervised algorithm, and the training data have only one classification. In this algorithm, a decision boundary is learned through the characteristics of normal samples, and this boundary is used to determine whether the new data point is similar to the training data; data beyond the boundary are regarded as abnormal data. Shawe-Taylor and Žličar [40] applied the OC-SVM to identify potential anomalies in financial time-series data and to find the distribution and the timing of the occurrence of the anomalous behavior in these data. The experiment results indicated that the OC-SVM detected changes in anomalous behavior in synthetic data sets and in several empirical data sets.

An SOM [42] is an unsupervised clustering algorithm. An SOM differs from other clustering algorithms in that it has a topological map that is used to express the distribution of each output or cluster. Therefore, SOMs can express the original high-dimensional space data through the visualized low-dimensional space, and the visualized result can also effectively explain the result of the grouping. Li *et al.* [41] used an SOM in a dynamic environment for discovering the abnormal financial behaviors of corporations. The experiment results indicated that the combination of macroeconomic indicators and financial indicators is superior to the use of financial indicators alone. The utilization of a hierarchical SOM helps in identifying the abnormal financial behaviors associated with corporate operations more effectively.

The results for the OC-SVM, SOM, and our proposed method are provided in Table 6. Our method outperformed the other methods in anomaly detection accuracy, and it exhibited the strongest detection capabilities in 10 rolling tests. To validate that our proposed model has more favorable detection capabilities, we used a t test. The t test is used to determine whether a significant difference exists between two averages. Before the statistical test, the population variance was unknown. We used the f test to verify that the populations had the same variance.

When the p values of the f test and t test are less than the significance level of 0.05, the null hypothesis can be rejected, which means the sampling was from different populations. In our results, the t test returned significant p values; thus, the null hypothesis was rejected. Therefore, the proposed method outperformed the control group. To obtain the value of each time period fairly, we ran each model 30 times. Tables 7 and 8 provide the comparison results of our method versus the OC-SVM and SOM, respectively. The t-test results

TABLE 6. Anomaly detection accuracy of different alternative models with S&P 100 constituent stocks data set.

Anomaly Detection	Rolling Test										Average
	1	2	3	4	5	6	7	8	9	10	
OCSVM	0.5509	0.5394	0.5442	0.5501	0.5439	0.5361	0.5508	0.5432	0.5652	0.5335	0.5457
SOM	0.5951	0.5306	0.5881	0.5949	0.6219	0.5598	0.5497	0.6349	0.6518	0.5803	0.5907
Proposed GACG	0.8212	0.7907	0.8093	0.8027	0.818	0.8006	0.7928	0.7978	0.8429	0.8368	**0.8113**

TABLE 7. Statistical test of proposed GACG and OCSVM.

	Rolling Test									
	1	2	3	4	5	6	7	8	9	10
F-value	47.70736	9.42140	36.30940	13.45678	33.73106	52.48977	23.38555	29.89704	38.24091	2.39872
P-value	4.20E-09	0.00326	1.24E-07	0.00053	2.82E-07	1.14E-09	1.01E-05	1.01E-06	6.77E-08	0.12687
Test result	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Accept H0
T-value	124.38467	413.67958	120.46406	113.82810	133.68709	112.32497	148.25440	149.31730	190.73750	429.87419
P-value	4.12E-72	2.44E-102	2.62E-71	6.92E-70	6.37E-74	1.49E-69	1.61E-76	1.06E-76	7.46E-83	2.63E-103
Test result	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0

TABLE 8. Statistical test of proposed GACG and SOM.

	Rolling Test									
	1	2	3	4	5	6	7	8	9	10
F-value	31.61851	9.08195	50.03186	97.66312	46.27081	11.23722	51.40315	77.79999	75.52385	28.86643
P-value	5.65E-07	0.00382	2.21E-09	4.80E-14	6.29E-09	1.42E-03	1.52E-09	2.63E-12	4.31E-12	1.43E-06
Test result	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0
T-value	84.07712	167.15383	40.20941	45.73598	69.99179	141.44962	38.64878	56.08296	51.86619	155.89619
P-value	2.66E-62	1.55E-79	4.68E-44	3.34E-47	9.98E-58	2.44E-75	4.29E-43	3.16E-52	2.69E-50	8.79E-78
Test result	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0	Reject H0

revealed that our model outperformed the control group in terms of detecting abnormalities.

E. INTERPRETATION OF DETECTED ANOMALIES

To verify the effectiveness of the proposed method for detecting anomalies, we used case studies to illustrate the anomalous time points captured by our model and the corresponding real events to prove the accuracy of the model in capturing anomalies. We want to emphasize that these results, which are based on unsupervised anomaly detection, only obstruct progress. Accidents and events differ, and they may appear suddenly without prior activity. The proposed method allows for detection of such events and related anomalies. We provide two examples of accidents to prove that our method can detect anomalous behavior in multiple time series data when an accident occurs.

a: Apple Inc. (AAPL)

iPhone Growth Concerns and Yuan Devaluation. 12 August 2015. Depicted in Figure 4, the GACG judges Apple's data on August 10, 2015, as anomalous (indicated by the red line), and the date of this anomalous activity is the Chinese stock market crash on August 12, 2015. This judgment also reflects the effect of early judgment rather than a reaction after the news was released. The global stock market crash set off by China caused a sharp drop in US stock markets. Even Apple Inc., which was preparing to launch a new product in September 2015, was strongly adversely affected, and its stock price plummeted. The slowdown in

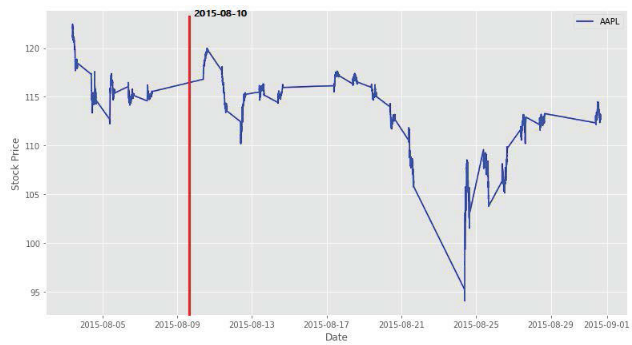


FIGURE 4. An anomalous event of Apple (AAPL) in the S&P 100 constituent stocks data set.

China's economy on August 12, 2015, became the main reason for the hit to Apple Inc.'s stock price. The stock market was concerned about the slowdown in Apple's growth. This is partly attributable to the slowdown in the growth of the Chinese market raising doubts about the demand for the new product, which put the stock price under pressure. When Apple's iPhone 6 and iPhone 6 Plus launched in 2014, they dominated the world in sales, recording the highest sales numbers ever. The key to success is the Chinese market, which accounts for one-fourth of the profit of global companies. However, as the Chinese economy weakened and consumer willingness to replace new devices decreased, Apple's stock price collapsed. This trend may continue and is likely to put pressure on Apple's China business. Although Apple also has substantial manufacturing operations in China, the

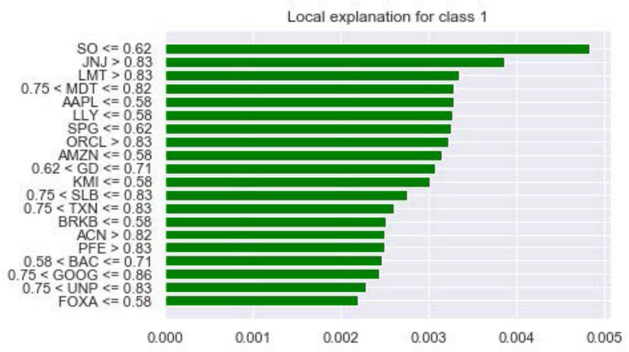


FIGURE 5. **Local explanation for an anomalous event of Apple (AAPL).**

FIGURE 6. **Anomaly event of Kinder Morgan (KMI) in S&P 100 constituent stocks data set.**

cost–benefit ratio from the devaluation of the Yuan is unlikely to be significant because Apple may calculate it in US dollars with its contract manufacturers. Even if the company has some influence over its contract manufacturers, the potential losses in revenue and growth in China may exceed any benefits. As shown in Figure 5, the Southern Company (SO) has the most influence on this decision, followed by Johnson & Johnson (JNJ) and Lockheed Martin Corporation (LMT).

b: Kinder Morgan, Inc. (KMI)

Slashes Dividend and Negative Headline. 04 December 2015. Figure 6 displays how the GACG judged Kinder Morgan's data on November 30, 2015, as anomalous (indicated by the red line), and the anomalous activity is the ''slashes dividend and negative headline'' published on December 4, 2015. Kinder Morgan's stock performance was slightly below analyst expectations. It caused some investors to worry that the company might be affected by the economic recession more than previously thought. After Kinder Morgan's poor performance in the second quarter of 2015, those worried investors began to feel genuine fear after the third quarter results were released in October. Although these results were within the expected range, the company announced on December 4, 2015, that it would reduce its dividend, and the negative news reports eventually caused disappointment among investors and caused the stock to plummet. As shown in Figure 7,Comcast Corporation (CMCSA) has the most influence on this decision, followed by Abbott Laboratories (ABT) and Visa Inc. (V).

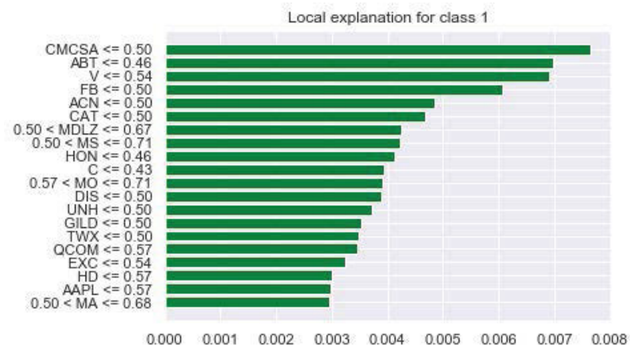


FIGURE 7. **Local explanation for anomaly event of Kinder Morgan (KMI).**

Based on the previously discussed description, these three cases demonstrate that the STCNN-RN combined with the GACG can discover anomalous activities that are forming. The model can detect anomalous activities early instead of at the time the news is released.

V. CONCLUSION

In this study, we focused on using the STCNN-RN combined with the GACG to discover anomalous time points with companies in the multiple financial time series data sets. Furthermore, we also introduced the LIME interpretable model to interpret and analyze the causes of anomalies.

First, we proposed the STCNN-RN to learn the complex correlation between multiple financial time series data sets. In the experiment, we used the S&P 100 Index Component stock data for the US stock market, the FTSE TWSE Taiwan 50 Index Component stock data set for the Taiwan stock market, and the SSE 50 Index Component stock data set for the Shanghai stock market. In these three data sets, the system has intraday transaction data and daily transaction data. Relative to the baseline models, our proposed model was able to capture the interaction between different financial products. The STCNN-RN model proved able to fit the data accurately in these three data sets. Compared with the multilayer DNN and TCNNRN, the STCNN-RN is more capable of processing data with temporal and spatial characteristics. Using different data sets, we proved the effectiveness of the STCNN-RN's model regardless of stock market, data frequency, or time interval. The model was proved to have a generalization ability for different data sets.

To catch the anomalous time points with companies and explain the anomaly phenomenon, we proposed the GACG and used this algorithm to discover the anomalous time points with companies where the model cannot be fitted among all transaction data. To prove the effectiveness of our proposed method, we also introduced baseline methods, including the Greedy method and the Random Choice method. From different models with different methods of selecting the best solution, we proved that the combination of the STCNN-RN and GACG can model the multiple financial time series data accurately and can also capture the anomalous time points with companies. Comprehensive analysis of the experiment results indicated that the STCNN-RN combined with GACG has a higher anomaly detection accuracy rate than other baseline models combined with different baseline methods in most of the rolling tests. At the same time, we also introduced our LIME interpretable model to interpret and analyze the results of our STCNN-RN combined with the GACG. Finally, we also verified the effectiveness of the STCNN-RN combined with the GACG to capture the anomaly through case studies.

To summarize the above description can be list as follows:

- The STCNN-RN and GACG can model the multiple financial time series data accurately and can also capture the anomalous time points with companies.
- Using different data sets, we proved the effectiveness of the STCNN-RN's model has a generalization ability for different data sets.
- The STCNN-RN combined with GACG has a higher anomaly detection accuracy rate than other baseline models in most of the rolling tests.
- We also introduced our LIME interpretable model to interpret and analyze the results of our STCNN-RN combined with the GACG.
- Finally, we also verified the effectiveness of the STCNN-RN combined with the GACG to capture the anomaly through case studies.

Future studies should improve some aspects of our model. Because this study explored the discovery of anomaly phenomena in all transaction data and the explanation of these abnormalities, investors can understand a stock market situation holistically. However, the data sets that our model learned were all without anomaly labels; this situation prevented our ensuring the effectiveness of the proposed model for catching all possible anomalies. Therefore, we can use some data set with labels to build models in future work. We can also use the STCNN-RN combined with a GA to formalize a prediction model, and the resultant model can remind investors of anomaly phenomena and reduce investment risks.

REFERENCES

[1] K. Alkhatib, H. Najadat, I. Hmeidi, and M. K. A. Shatnawi, "Stock price prediction using K-nearest neighbor (KNN) algorithm," *Int. J. Bus., Hum. Technol.*, vol. 3, no. 3, pp. 32–44, 2013.

[2] W. Fenghua, X. Jihong, H. Zhifang, and G. Xu, "Stock price prediction based on SSA and SVM," *Procedia Comput. Sci.*, vol. 31, pp. 625–631, 2014.

[3] O. Hegazy, O. S. Soliman, and M. A. Salam, "A machine learning model for stock market prediction," 2014, *arXiv:1402.7351*. [Online]. Available: https://arxiv.org/abs/1402.7351

[4] A. Kazem, E. Sharifi, F. K. Hussain, M. Saberi, and O. K. Hussain, "Support vector regression with chaos-based firefly algorithm for stock market price forecasting," *Appl. Soft Comput.*, vol. 13, no. 2, pp. 947–958, Feb. 2013.

[5] Y.-Y. Chen, C.-T. Chen, C.-Y. Sang, Y.-C. Yang, and S.-H. Huang, "Adversarial attacks against reinforcement learning-based portfolio management strategy," *IEEE Access*, vol. 9, pp. 50667–50685, 2021.

[6] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS ONE*, vol. 12, no. 7, Jul. 2017, Art. no. e0180944.

[7] R. Hafezi, J. Shahrabi, and E. Hadavandi, "A bat-neural network multi-agent system (BNNMAS) for stock price prediction: Case study of DAX stock price," *Appl. Soft Comput.*, vol. 29, pp. 196–210, Apr. 2015.

[8] K. Khare, O. Darekar, P. Gupta, and V. Z. Attar, "Short term stock price prediction using deep learning," in *Proc. 2nd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, May 2017, pp. 482–486.

[9] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1643–1647.

[10] A. A. Adebiyi, A. O. Adewumi, and C. K. Ayo, "Comparison of ARIMA and artificial neural networks models for stock price prediction," *J. Appl. Math.*, vol. 2014, pp. 1–7, Mar. 2014.

[11] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, "Stock price prediction using the ARIMA model," in *Proc. UKSim-AMSS 16th Int. Conf. Comput. Model. Simul.*, Mar. 2014, pp. 106–112.

[12] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4967–4976.

[13] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 18–32, Jan. 2014.

[14] P.-Y. Wang, C.-T. Chen, J.-W. Su, T.-Y. Wang, and S.-H. Huang, "Deep learning model for house price prediction using heterogeneous data analysis along with joint self-attention mechanism," *IEEE Access*, vol. 9, pp. 55244–55259, 2021.

[15] A. Lazarevic, L. Ertoz, V. Kumar, A. Ozgur, and J. Srivastava, "A comparative study of anomaly detection schemes in network intrusion detection," in *Proc. SIAM Int. Conf. Data Mining*, May 2003, pp. 25–36.

[16] H. Kwon and N. M. Nasrabadi, "Kernel RX-algorithm: A nonlinear anomaly detector for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 2, pp. 388–397, Feb. 2005.

[17] W. F. Sharpe, "Capital asset prices: A theory of market equilibrium under conditions of risk," *J. Finance*, vol. 19, no. 3, pp. 425–442, Sep. 1964.

[18] E. F. Fama and K. R. French, "Common risk factors in the returns on stocks and bonds," *J. Financial Econ.*, vol. 33, no. 1, pp. 3–56, Feb. 1993.

[19] A. Roy, J. Sun, R. Mahoney, L. Alonzi, S. Adams, and P. Beling, "Deep learning detecting fraud in credit card transactions," in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2018, pp. 129–134.

[20] M. Schreyer, T. Sattarov, D. Borth, A. Dengel, and B. Reimer, "Detection of anomalies in large scale accounting data using deep autoencoder networks," 2017, *arXiv:1709.05254*. [Online]. Available: https://arxiv.org/abs/1709.05254

[21] D. Huang, D. Mu, L. Yang, and X. Cai, "CoDetect: Financial fraud detection with anomaly feature detection," *IEEE Access*, vol. 6, pp. 19161–19174, 2018.

[22] S. Thiprungsri and M. Vasarhelyi, "Cluster analysis for anomaly detection in accounting data: An audit approach," *Int. J. Digit. Accounting Res.*, vol. 11, pp. 1–16, Jul. 2011.

[23] WRDS. (2015). *Wharton Research Data Services*. [Online]. Available: https://wrds-www.wharton.upenn.edu/

[24] C.-H. Kuo, C.-T. Chen, S.-J. Lin, and S.-H. Huang, "Improving generalization in reinforcement learning–based trading by using a generative adversarial market model," *IEEE Access*, vol. 9, pp. 50738–50754, 2021.

[25] B. Podobnik and H. E. Stanley, "Detrended cross-correlation analysis: A new method for analyzing two nonstationary time series," *Phys. Rev. Lett.*, vol. 100, no. 8, Feb. 2008, Art. no. 084102.

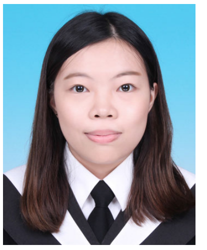
[26] Y. Wang, Y. Wei, and C. Wu, "Cross-correlations between chinese A-share and B-share markets," *Phys. A, Stat. Mech. Appl.*, vol. 389, no. 23, pp. 5468–5478, Dec. 2010.

[27] L. Liu, "Cross-correlations between crude oil and agricultural commodity markets," *Phys. A, Stat. Mech. Appl.*, vol. 395, pp. 293–302, Feb. 2014.

[28] L. Kullmann, J. Kertész, and K. Kaski, "Time-dependent cross-correlations between different stock returns: A directed network of influence," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 66, no. 2, Aug. 2002, Art. no. 026125, doi: 10.1103/PhysRevE.66.026125.

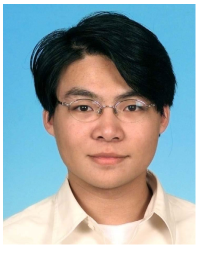
[29] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[30] Y. Hua, L. Mou, and X. X. Zhu, "Relation network for multilabel aerial image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 7, pp. 4558–4572, Jul. 2020.
[31] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2425–2433.
[32] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-based anomaly detection," 2018, *arXiv:1802.06222*. [Online]. Available: https://arxiv.org/abs/1802.06222
[33] T. Leangarun, P. Tangamchit, and S. Thajchayapong, "Stock price manipulation detection using generative adversarial networks," in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, Nov. 2018, pp. 2104–2111.
[34] Q. Wang, W. Xu, X. Huang, and K. Yang, "Enhancing intraday stock price manipulation detection by leveraging recurrent neural networks with ensemble learning," *Neurocomputing*, vol. 347, pp. 46–58, Jun. 2019.
[35] S. R. Islam, S. K. Ghafoor, and W. Eberle, "Mining illegal insider trading of stocks: A proactive approach," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 1397–1406.
[36] D. Gunning, "Explainable artificial intelligence (XAI)," *Defense Adv. Res. Projects Agency (DARPA)*, vol. 2, p. 2, Nov. 2017.
[37] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?' explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.
[38] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, Apr. 2018, pp. 1–9.
[39] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti, "Local rule-based explanations of black box decision systems," 2018, *arXiv:1805.10820*. [Online]. Available: https://arxiv.org/abs/1805.10820
[40] J. Shawe-Taylor and B. Žličar, "Novelty detection with one-class support vector machines," in *Advances in Statistical Models for Data Analysis*. Cham, Switzerland: Springer, 2015, pp. 231–257.
[41] S.-C. Li, C.-F. Huang, C.-C. Tu, and A.-P. Chen, "Discovery of abnormal financial behavior in a dynamic finance environment with hierarchical self-organizing mapping," in *Proc. 2nd Int. Conf. Softw. Eng. Data Mining*, Jun. 2010, pp. 450–455.
[42] T. Kohonen, *Cybernetic Systems: Recognition, Learning, Self-Organization*, E. R. Caianiello and G. Musso, Eds. Letchworth, U.K.: Research Studies Press, 1984, p. 3.



MEI-SEE CHEONG received the B.S. degree in information management from the National Taiwan University of Science and Technology, Taipei, Taiwan, in 2018, and the M.S. degree in information management from National Chiao Tung University, Hsinchu, Taiwan, in 2020. Her research interests include relational learning and explainable AI.

MEI-CHEN WU received the B.S. and M.S. degrees in information management from the Yu Da University of Science and Technology, Miaoli, Taiwan, in 2012 and 2013, respectively. She is currently pursuing the Ph.D. degree in information management with National Chiao Tung University, Hsinchu, Taiwan. Her research interests include information hiding, digital watermarking, deep learning, artificial intelligence, and financial technology.



SZU-HAO HUANG (Member, IEEE) received the B.E. and Ph.D. degrees in computer science from National Tsing Hua University, Hsinchu, Taiwan, in 2001 and 2009, respectively. He is currently an Assistant Professor with the Department of Information Management and Finance and the Chief Director with Financial Tehnology (FinTech) Innovation Research Center, National Yang Ming Chiao Tung University, Hsinchu. He is also the Principal Investigator of the MOST Financial Technology Innovation Industrial-Academic Alliance and several cooperation projects with leading companies in Taiwan. He has authored more than 50 papers published in the related international journals and conferences. His research interests include artificial intelligence, deep learning, recommender systems, computer vision, and financial technology.