foods

MDPI

Article

CSGNN: Contamination Warning and Control of Food Quality via Contrastive Self-Supervised Learning-Based Graph Neural Network

Junyi Yan [1], Hongyi Li [2], Enguang Zuo [3,*], Tianle Li [3], Chen Chen [3], Cheng Chen [1] and Xiaoyi Lv [1,*]

[1] College of Software, Xinjiang University, Urumqi 830046, China
[2] Guangzhou Panyu Polytechnic, No. 1342 Shiliang Road, Guangzhou 511483, China
[3] College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China
[*] Correspondence: zeg@stu.xju.edu.cn (E.Z.); xjuwawj01@163.com (X.L.)

Abstract: Effective contamination warning and control of food quality can significantly reduce the likelihood of food quality safety incidents. Existing food contamination warning models for food quality rely on supervised learning, do not model the complex feature associations between detection samples, and do not consider the unevenness of detection data categories. In this paper, To overcome these limitations, we propose a Contrastive Self-supervised learning-based Graph Neural Network framework (CSGNN) for contamination warning of food quality. Specifically, we structure the graph for detecting correlations between samples and then define the positive and negative instance pairs for contrastive learning based on attribute networks. Further, we use a self-supervised approach to capture the complex relationships between detection samples. Finally, we assessed each sample's contamination level based on the absolute value of the subtraction of the prediction scores from multiple rounds of positive and negative instances obtained by the CSGNN. Moreover, we conducted a sample study on a batch of dairy product detection data in a Chinese province. The experimental results show that CSGNN outperforms other baseline models in contamination assessment of food quality, with AUC and recall of unqualified samples reaching 0.9188 and 1.0000, respectively. Meanwhile, our framework provides interpretable contamination classification for food detection. This study provides an efficient early warning method with precise and hierarchical contamination classification for contamination warning of food quality work.

Keywords: food quality safety; contamination warning; contrastive learning; self-supervised learning; graph neural networks

check for updates

Citation: Yan, J.; Li, H.; Zuo, E.; Li, T.; Chen, C.; Chen, C.; Lv, X. CSGNN: Contamination Warning and Control of Food Quality via Contrastive Self-Supervised Learning-Based Graph Neural Network. *Foods* **2023**, *12*, 1048. https://doi.org/10.3390/foods12051048

Academic Editor: Rosario Linacero

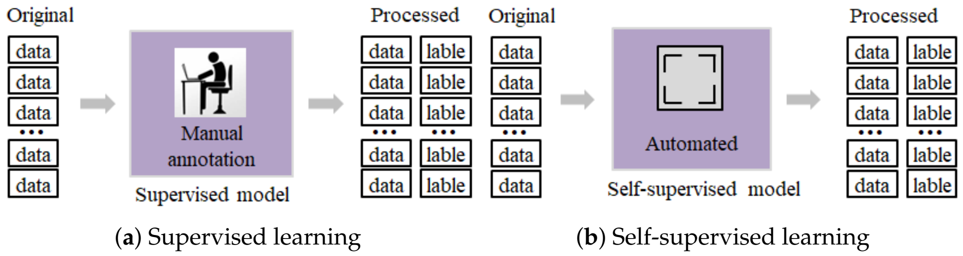
Received: 24 November 2022
Revised: 17 February 2023
Accepted: 24 February 2023
Published: 1 March 2023

1. Introduction

Food safety issues are of increasing concern to international organizations and people worldwide. Efficient surveillance and early warning programs can effectively reduce the probability of food safety accidents. Currently, many international organizations and countries have established monitoring systems to ensure food quality safety [1–4]. Similarly, China has gradually improved its national food safety risk assessment system [5,6]. For example, in 2009, China enacted the Food Safety Law of the People's Republic of China. In 2011, the China National Center for Food Safety Risk Assessment (CFSA) was established. In 2018, the revised version of the Food Safety Law of the People's Republic of China included food safety assessment as a scientific basis for implementing regulations and setting standards. Therefore, developing food safety risk assessment methods can help systematize and standardize China's food safety regulatory system. Contamination of food quality is one of the significant causes of food safety risks. Contamination warning and control of food quality are closely related to food safety risk assessment and are an essential part of food safety regulations.

The existing mainstream contamination warning methods mainly include hierarchical relationship analysis-based methods [7–9], Bayesian network-based methods [10,11], and artificial neural network (ANN)-based methods [12]. However, these approaches exhibit the following deficiencies.

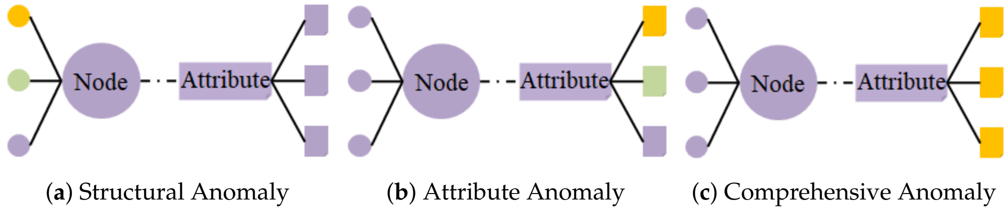
1. They rely on supervised learning [13]. Still, manual labeling of detection data labels will significantly increase the time cost and require operators to have an explicit knowledge of data category classification. Once a casual error in classifying data categories occurs, it will lead to a series of subsequent tasks with persistent interference from subjective factors, which is fatal in practical application scenarios. The supervised learning process on raw data is shown in (a) in Figure 1.
2. They use balanced training data or do not consider the category imbalance in the training data. Data category imbalance is a significant quantitative difference in the sample size of different labels in the data, which is common in practical scenarios. Category imbalance can limit the model's performance to varying degrees [14,15], so it is critical to investigate how to adopt strategies to address the data category imbalance while ensuring relatively good performance [16].
3. They do not adequately capture topological information between detection samples. The data obtained in the detection process has the characteristics of complexity, nonlinearity and discreteness, which means that we need to pay attention to the detection data's attribute information and topology information as much as possible to realize the contamination warning of food quality more accurately.



(**a**) Supervised learning (**b**) Self-supervised learning

Figure 1. Supervised learning versus self-supervised learning.

Contrastive learning is a promising solution to the above limitations. Contrastive learning uses a self-supervised approach to construct supervised information from the data, essentially addressing the reliance on manual labeling [17], processed as in (b) in Figure 1. Contrastive learning focuses on learning common features between instances of the same class and distinguishing differences between different classes by modeling the relationship between each node and some of its adjacent substructures [18]. Moreover, contrastive learning has powerful advantages in graph representation learning [19,20], especially for anomaly detection in attribute networks [21]. The learned embeddings in attribute networks include attribute and structure information, effectively capturing topological and attribute information in the network [22–24]. Figure 2 shows three anomalies that attribute networks are committed to capturing. The contamination warning of food quality task aims to mine all quality-contaminated unqualified samples and potentially contaminated qualified samples, that is, to discover anomalous samples whose characteristic information differs from most normal samples, which is similar to the principle of anomaly detection in attribute networks [25]. Graph neural networks (GNNs) model complex correlations among sample individuals. Therefore, attribute network-based contrastive learning has the potential to be applied to the contamination warning of food quality tasks.



(**a**) Structural Anomaly (**b**) Attribute Anomaly (**c**) Comprehensive Anomaly

Figure 2. The attribute network catches three exceptions. Subgraph (**a**) belongs to the structural exception. That is, there are connection nodes that do not match all properties. Subgraph (**b**) belongs to the attribute exception. That is, some attributes do not match all nodes. Subgraph (**c**) belongs to the comprehensive exception. That is, there are both structural exceptions and attribute exceptions.

Based on the above observations, we propose a novel Contrastive Self-supervised learning-based Graph Neural Network framework (CSGNN for abbreviation) for contamination warning and control of food quality. Specifically, we first construct the food detection data into an attribute graph containing attributes and structural information [26,27]. Further, the self-supervised contrastive learning module is trained by sampling the positive and negative instance pairs obtained from the complete attribute graph. Therefore, CSGNN can efficiently exploit detection data's structural information and contrastive association information to accomplish contamination warning and control of food quality. In brief, the main contributions of the food safety contamination early warning framework proposed in this paper are as follows.

1. An end-to-end contamination warning and control framework for food quality is proposed by us, which can efficiently mine quality-contaminated unqualified samples and potentially contaminated qualified samples in food detection data by obtaining contamination values from food detection data.
2. A contrastive self-supervised learning scheme for contamination warning and control of food quality is proposed by us. Contrastive learning solves the dependence of the previous method on the balance of data categories. Meanwhile, self-supervised learning efficiently solves the problems of being easily interfered with by subjective factors and high time costs in the manual labeling operation of data categories in practical applications.
3. GNNs are used for information transfer. The CSGNN considers the data nodes' attributes and structural information by constructing an attribute graph. To the best of our knowledge, it is also the first time graph algorithms have been applied to food safety risk assessment-related tasks.
4. The data in the actual scenarios verify that the contamination warning effect of the proposed algorithm is better than that of the current mainstream model. On a batch of dairy product detection data in a specific province in China, we compared the CSGNN framework with the mainstream model. Under self-supervision, the recall of unqualified samples of CSGNN reached 1.0000, which was more than 13% higher than the sub-optimal model. In addition, CSGNN completed the contamination classification of the food detection data based on the contamination value of each sample.

2. Related Work

2.1. Contamination Assessment Models for Food Quality

There are many types of traditional contamination assessment models for food quality. Various assessment methods exhibit different performances. Specifically, back-propagation (BP) may fall into an optimal local solution and lead to training failure during the training process [28]. The support vector machine (SVM) can guarantee to find the optimal global resolution with the help of convex optimization, which has been applied in various detection tasks [29,30]. Still, it cannot fully exploit the potential risks of food safety data. Bayesian network models have also been applied to contamination assessment tasks for food quality [31,32]. Still, they usually require modeling with expert knowledge, resulting in model performance limited by specialist experience.

With the development of deep learning, deep neural network models (DNNs) have shown promising potential in mining data features [33,34], which provides new ideas for contamination assessment for food quality. Nogales et al. applied multilayer perceptron (MLP) and one-dimensional convolutional neural network (Conv1D) to food quality contamination prediction workflow for the RASFF data launched by European Union, combining entity embedding to obtain better prediction accuracy than machine learning models [35]. Geng et al. used deep radial basis function (DRBF) neural networks combined with hierarchical analysis (AHP) to assess the contamination of food detection data [8], which enhanced the data representation of RBF shallow networks while avoiding getting trapped in local optima. However, these models are highly influenced by subjective metrics and cannot adequately capture the correlation information among complex detection data. In contrast, this paper proposes for the first time to use GNN to model the samples directly and the association information between them [36] to mine the association information in the detection data, ultimately, to achieve a more precise contamination assessment for food quality.

2.2. Contrastive Learning

Contrastive learning is an essential part of self-supervised learning. It completes the representation of data features by constructing a pair of instances and feeding them into the contrastive learning module. Contrastive objects and contrastive losses highlight inconsistencies between the different classes and similarity features between the same classes, respectively, which align with the original goals of downstream tasks such as classification and detection [37].

With the development of GNN, contrastive learning has also been applied to the training of GNN. DGI captures the global structural information in the network by maximizing the interaction information between the local and global input [19]. GraphCL learns node embeddings by maximizing the representational similarity between intrinsic features and link structures for local subgraphs of the same node [38]. Through a contrastive loss, SUBLIME maximized the mutual information between anchor maps and learned structure maps [39]. Although contrastive self-supervised learning has improved the performance of GNNs in speech recognition and visual representation learning [40,41], to the best of our knowledge, the current study is the first to propose its use as a contamination assessment model for food quality. Due to the particularity of the contamination assessment task, we designed the contrastive learning module to focus on the local information of the data rather than the global information, which will help the model to mine the data features of contaminated samples more efficiently.

3. Materials and Methods

In this paper, we use bold lowercase letters (e.g., $\mathbf{X}$), bold uppercase letters (e.g., X), and fancy letters (e.g., $\mathcal{G}$) to denote vectors, matrices, and sets, respectively. The main symbols used in this paper are summarized in Table 1.

3.1. Problem Definition and Data Source

3.1.1. Problem Definition

Given a set of food detection data with N samples and V testing indicators $x_{v1,\dots,}x_{vn}$, first construct an attribute graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ for the detection data, where $\mathcal{V}$ is the set of nodes of $\mathcal{G}$ ($|\mathcal{V}| = N = n$), $\mathcal{E}$ is the set of edges of $\mathcal{G}$, and $\mathbf{X} \in \mathbb{R}^{n \times d}$ is the attribute matrix of $\mathcal{G}$ ($d = V$). The objective is to compute the contamination value $f(s_i)$ for each sampled sample s_i (a higher contamination value means higher hazardous contamination of the sample presence). By ranking the contamination values of all samples, the model is based on the lowest contamination value W of the unqualified samples and the more obvious boundary value U between the contamination values of the contaminated samples and the negligible contaminated samples to achieve the contamination ranking of the detection data.

Table 1. Notes and explanations related to the CSGNN framework. The table's three blocks (from top to bottom) show the data preprocessing and structured representation, GCN-based contrastive learning, and CSGNN's hyperparameters, respectively.

Notation	Description and Explanation
x_v^{min}	The minimum value of the original value of the v-th indicator for all samples.
x_v^{max}	The maximum value of the original value of the v-th indicator for all samples.
x_v^{mean}	The average value of the original value of the v-th indicator for all samples.
x_{vn}	The original value of the v-th indicator of the n-th sample.
N	The number of samples in the detection data.
V	The number of testing indicators in the detection data.
n	The number of nodes in $\mathcal{G}$.
d	The dimension of the attribute in $\mathcal{G}$.
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$	Attribute networks constructed from the detection data.
$\mathcal{V}$	The set of nodes of $\mathcal{G}$.
$\mathcal{E}$	The set of edges of $\mathcal{G}$.
$\mathbf{X} \in \mathbb{R}^{n \times d}$	The attribute matrix of $\mathcal{G}$.
$I_i = (s_i, \mathcal{G}_i, y_i)$	Instance pairs for each batch with a total batch size of M.
s_i	Sampled samples in instance pair I_i.
$\mathcal{G}_i$	The group of adjacent samples in instance pair I_i.
$y_i \in \{0, 1\}$	True label of the sampled sample s_i.
$\mathbf{H}_i^{(\ell)} \in \mathbb{R}^{\mathbf{a} \times b^{(\ell)}}$	The representation matrix is learned by the ℓ-th implicit layer.
$\mathbf{W}^{(\ell)} \in \mathbb{R}^{d^{(\ell)} \times d^{(\ell+1)}}$	The ℓ-th layer trainable weight matrix.
$\mathbf{E}_i \in \mathbb{R}^{a \times b}$	The embedding matrix of the nodes in $\mathcal{G}_i$.
$\mathbf{z}_i^{(\ell)} \in \mathbb{R}^{b^{(\ell)}}$	The row vector of feature representations of the sampled sample s_i learned by the ℓ-th implicit layer.
$\mathbf{e}_i^{ss} \in \mathbb{R}^b$	The embedding vector of s_i.
$\mathbf{e}_i^{as} \in \mathbb{R}^b$	The embedding vector of $\mathcal{G}_i$.
$\mathbf{W}^{(b)} \in \mathbb{R}^{b \times b}$	The weight matrix of the comparison recognition module.
p_i	The prediction score of I_i.
$f(s_i) \in [0, 1]$	The contamination value of the sampled sample s_i.
R	The number of sampling rounds.
a	The number of nodes in adjacent sample groups.
b	The dimensionality of embedding.
W	The lowest contamination value of the unqualified samples.
U	The more obvious boundary value between the contaminated sample and the negligible contamination sample (default 0.5).
Z	Set the number of edges when structuring.

3.1.2. Data Source

In this paper, 2158 detection data of sterilized dairy products(qualified samples: unqualified samples = 2117:41) from 2013 to 2021 provided by the Institute of Product Quality Supervision and Inspection in a Chinese province, were used to test the contamination assessment capability for food quality of the models. According to the National Food Safety Standard of China [42], the testing indicators of sterilized dairy products include five categories: sensory indicators, physical and chemical indicators, contaminant index, mycotoxin index and microbial index. The selection of contamination assessment for food quality testing indicators should be considered comprehensively in terms of operability and validity. According to the obtained detection data, we aim to scientifically select the testing indicators corresponding to the factors that may cause food quality contamination. Since the microbial index in the detection data has met the requirements [8], we selected the physicochemical index and mycotoxin index from the testing indicators of sterilized milk specified in the national standards as the evaluation criteria for food quality contamination, where the physicochemical index includes lactose, nonfat milk solids, protein, acidity and fat for a total of five indicators, In addition, the mycotoxin index refers to aflatoxin M_1. Table 2 shows the six testing indicators' specific requirements and testing methods. And Table 3 presents detection data for partially sterilized dairy products.

Table 2. The specific requirements of the six testing indicators and testing methods [1].

Item		Requirements	Testing Method
physicochemical index	Lactose/(g/100 g)	⩽2.0	GB 5009.8-2016
	Protein/(g/100 g)	⩾3.1	GB 5009.5-2010
	Acidity/(°T)	11∼16	GB 5413.34-2010
	Fat/(g/100 g)	⩾3.7	GB 5413.3-2010
	Nonfat Milk Solids/(g/100 g)	⩾8.5	GB 5413.39-2010
mycotoxin index	Aflatoxin M_1/(μg/kg)	⩽0.5	GB 2761-2017

[1] http://down.foodmate.net/standard/index.html (accessed date: 10 August 2022)

Table 3. Detection data for partially sterilized dairy products from 2013 to 2021.

Sample Number	Testing Date	Item					
		Lactose	Nonfat Milk Solids	Protein	Acidity	Aflatoxin M_1	Fat
20211010-578	10 October 2021	1.73	8.97	3.40	12.00	0.2	4.28
20200410-525	10 April 2020	1.74	8.67	3.39	12.08	0.5	4.69
20190504-166	4 May 2019	1.72	8.81	3.37	12.07	0.5	4.44
20180610-453	10 June 2018	1.70	8.68	3.25	12.19	0.5	4.36
20210909-512	9 September 2021	1.73	8.62	3.42	12.40	0.2	4.20

3.2. Contamination Warning of Food Quality Based on Contrastive Self-Supervised Learning

In this section, we describe the overall framework of CSGNN, as shown in Figure 3. The CSGNN framework consists of four parts: data preprocessing and structuring, contrastive instance pair sampling, GCN-based contrastive learning, and contamination assessment. First, the raw data are preprocessed and structured better to implement feature mining for detection data with complex correlations. Next, the local subgraph sampling strategy of RWR is used to obtain positive and negative instance pairs. After that, we use the GCN-based contrastive learning model to train each batch of instance pairs. Finally, all the detection samples will be randomly traversed as sampling samples. The contamination assessment of the detection samples is completed by evaluating the consistency between the sampling samples and the adjacent sample groups.

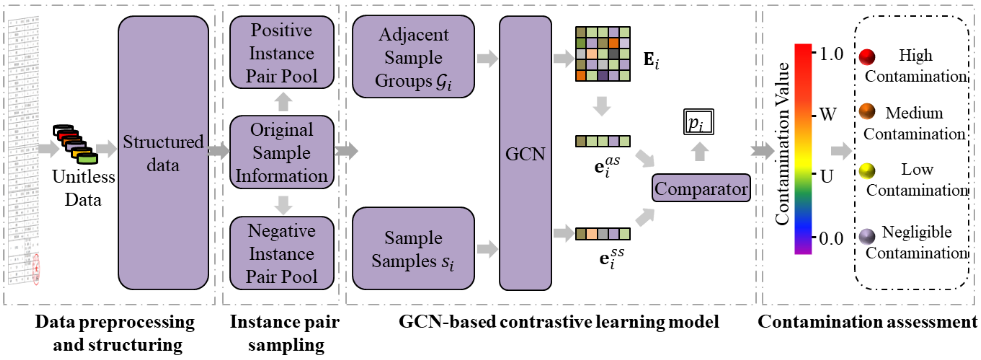


Figure 3. The overall framework of CSGNN. First, the framework preprocesses and structures the raw detection data to better mine the complex relationships in the detection data. Then, the local subgraph sampling strategy is used to obtain positive and negative instance pairs, which are trained by the GCN-based contrastive learning model and get the contamination value of each sample. Finally, the contamination assessment for food quality is carried out by the contamination value of the sample.

The food quality hazard contamination considered in this paper refers to the contamination of food quality due to testing indicators that do not meet the standard requirements. CSGNN is based on contrastive learning to measure the consistency between the performance of the sampled samples and their adjacent sample groups in each testing indicator. When the samples show a higher degree of inconsistency with the adjacent sample groups, their contamination values are higher, and the possibility of quality hazard contamination is more significant.

3.2.1. Data Preprocessing and Structuring

We visualized part of the raw detection data for sterilized dairy products in Table 3, as shown in Figure 4. It can be seen from the figure that there are significant dimensional differences between the different testing indicators.

As is demonstrated in the first part of Figure 3, we used the minimum-maximum normalization method to transform the raw data into unitless data to eliminate the dimensional differences among different testing indicators. According to the different requirements of the food safety standards for the six testing indicators, we classified them into three categories: positive indicators, inverse indicators, and oscillatory indicators, as shown in Table 4. The positive indicator is one whose hazardous contamination increases as the indicator's value increases. Conversely, the inverse indicator is one whose hazardous contamination decreases as the indicator's value increases. In addition, the closer the oscillator is to the specified interval, the less contamination there is, and the further it is from that interval, the more contamination there is. Equations (1)–(3) normalize the three categories of indicators, respectively. Ultimately, the greater the normalized data value, the greater its hazardous contamination.

$$x_{vn}^{p} = \frac{x_{vn} - x_v^{\text{min}}}{x_v^{\text{max}} - x_v^{\text{min}}} \tag{1}$$

$$x_{vn}^{r} = 1 - \frac{x_{vn} - x_v^{\text{min}}}{x_v^{\text{max}} - x_v^{\text{min}}} = \frac{x_v^{\text{max}} - x_{vn}}{x_v^{\text{max}} - x_v^{\text{min}}} \tag{2}$$

$$x_{vn}^{o} = \frac{|x_{vn} - x_v^{\text{mean}}|}{x_v^{max} - x_v^{\text{min}}} \tag{3}$$

where, $x_v^{\text{max}} = \max\{x_{v1}, x_{v2}, \ldots, x_{vn}\}$, $x_v^{\text{min}} = \min\{x_{v1}, x_{v2}, \ldots, x_{vn}\}$, $x_v^{\text{mean}} = \frac{\sum_{i=1}^{n} x_{vi}}{n}$. x_{vi} denotes the value of the v-th indicator corresponding to the i-th sample, and n denotes the number of samples.

Table 4. Sterilized dairy products testing indicators categories classification.

Categories	Item	Requirements
Positive indicators	Aflatoxin M_1/(μg/kg)	⩽0.5
	Lactose/(g/100 g)	⩽2.0
Inverse indicators	Protein/(g/100 g)	⩾3.1
	Fat/(g/100 g)	⩾3.7
	Nonfat Milk Solids/(g/100 g)	⩾8.5
Oscillatory indicators	Acidity/(°T)	11∼16

We construct a representation based on the correlations between detection samples. Expressly, we represent the food detection samples as nodes in the graph. The testing indicators of the samples are represented as node attributes in the graph. Accordingly, the construction of the attribute graph is completed. The distances between each sample are calculated separately and arranged in descending order when constructing the sterilized milk detection data graph. Our experiments show that the model performs best comprehensively when the top Z = 50 samples with the closest distance are set as edged, and the rest are set as unedged. CSGNN is based on self-supervised learning to detect and analyze all samples in a dataset. It uses preprocessing and structuring operations to process the original complex and discrete detection data into the structured representation suitable for GNN. To demonstrate the impact of preprocessing and structuring operations on the initial detection data, we visualized 100 samples (randomly selected) as an example. Figure 5 shows the reduced dimensional distribution of the sterilized milk data before preprocessing and the network structure after preprocessing and structuring, respectively. Our processing is well suited to transform the original detection data between different magnitudes into structured data suitable for GNN without losing as much valid feature information and topological relationships as possible.

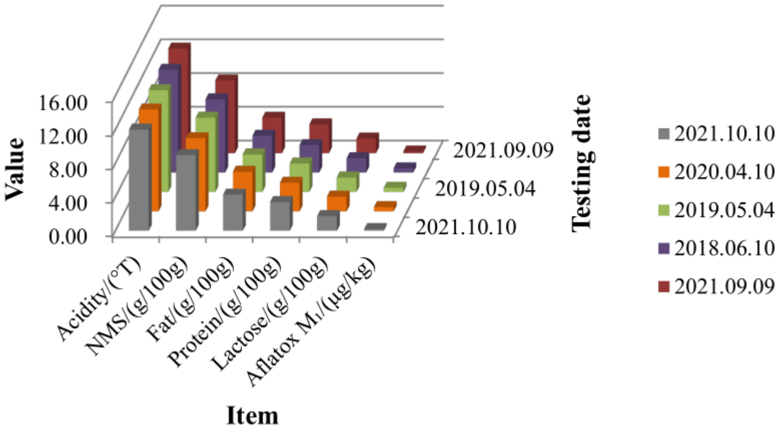
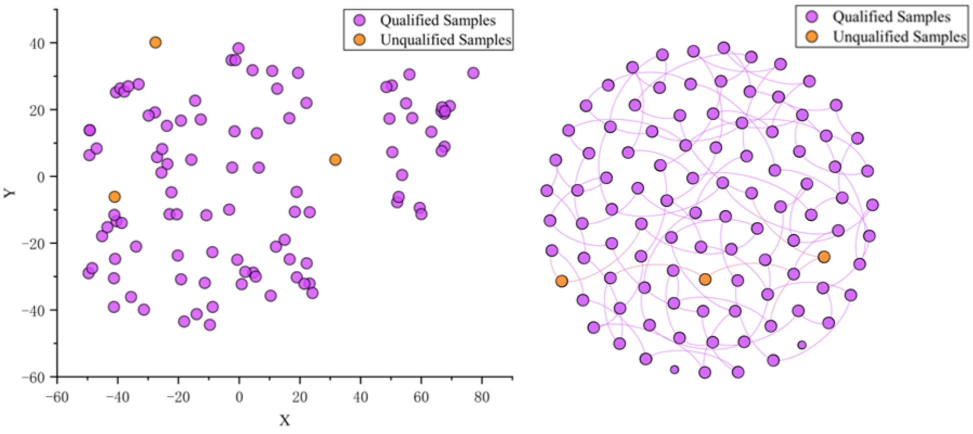


Figure 4. Visualization of raw partial detection data values for some sterilized dairy products.



(**a**) (**b**)

Figure 5. Effect of preprocessing and structuring operations on the original detection data (t-SNE visualization with 100 randomly selected samples). Subfigure (**a**) shows the dimensionality reduction distribution of the sterilized milk detection data before preprocessing. Subfigure (**b**) shows the result of preprocessing and structuring the original data with node diameters proportional to PageRank scores, coloring the node according to sample classes and the edge colors determined by the source node colors. (**a**) Before data preprocessing. (**b**) After data preprocessing and structuring.

3.2.2. Contrastive Instance Pair Sampling

The definition of contrastive instance pairs is the core work of the contrastive learning module. Previous works have demonstrated different advantages in defining instance pairs of graphs [20,43]. Due to the complex topological relationships between different samples of food detection data, we expect the contamination assessment framework for food quality to capture both attribute and structural information of the samples in an integrated manner. This part corresponds to the second part of the content of Figure 3. Inspired by the novel contrastive instance pairs designed for anomaly detection tasks in the paper [21], we focus on modeling the relationships between target nodes and their neighboring subgraphs to help mine the local information of the nodes. Specifically, we use a "sampled samples v.s. adjacent sample groups" instance pair for the attribute network in the CSGNN framework. The first element of the instance pair is a random sample obtained by traversal in the detection data. The second element of the instance pair refers to the group of adjacent samples sampled from the initial samples. The initial sample is set to the sampled sample for positive instance pairs. That means the sampled adjacent sample group matches the nearby samples of the sampled sample. For negative instance pairs, the initial sample is drawn randomly from all samples that do not include the sampled sample; that is, the initial sample will not come from the sampled sample. Therefore, there is a certain degree of mismatch between their sampled samples and adjacent sample groups for contaminated samples. A higher degree of mismatch represents higher hazardous contamination of the detection sample corresponding to that node.

Figure 6 shows the sampling process of instance pairs in the CSGNN framework. Sampling consists of selecting the sampled samples, sampling adjacent sample groups, hiding sampled samples, and synthesizing instance pairs.

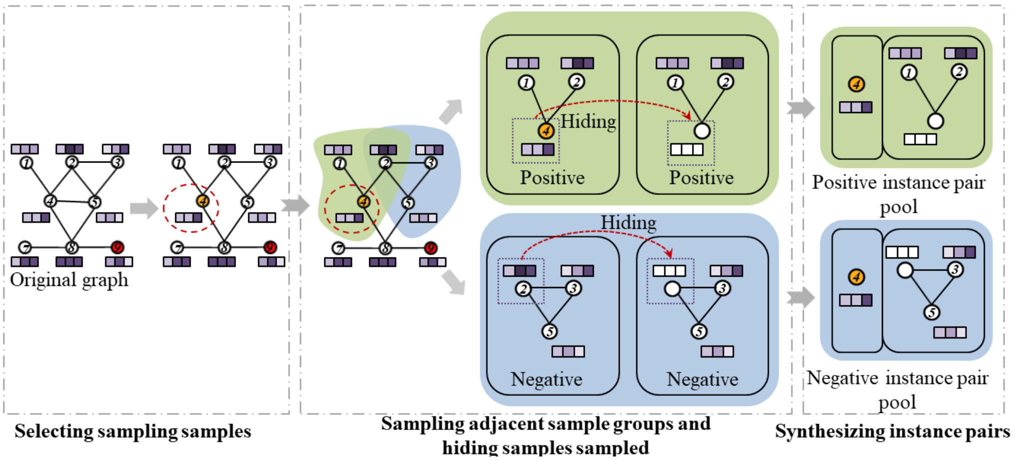

Original graph
Hiding
Positive
Positive
Hiding
Negative
Negative
Positive instance pair pool
Negative instance pair pool
Selecting sampling samples
Sampling adjacent sample groups and hiding samples sampled
Synthesizing instance pairs

Figure 6. Contrastive learning completes the instance pair sampling process. Node 4 is the sampled sample, and node 9 is the unqualified sample.

1. Selecting the sampled samples. All samples in the detection data are traversed randomly within each epoch, and the sampling samples are determined randomly.
2. Sampling adjacent sample groups. We set their initial samples as sampling samples and random sampling samples for the adjacent sample groups of positive and negative instance pairs, respectively. Inspired by the paper [43], we use RWR [44] as the sampling strategy for local sample groups to make the sampling strategy for adjacent sample groups more efficient.
3. Hiding sampled samples. To avoid the contrastive learning module to quickly identify the presence of sampled samples in the adjacent sample groups, we zeroed out the attribute features of the initial samples. That is, the attribute information of the sampled samples is hidden.
4. Synthesizing instance pairs. Combine sampled and adjacent samples into instance pairs and save them to the positive and negative instance pairs sample pool, respectively.

3.2.3. GCN-Based Contrastive Learning

GNNs use information propagation between nodes to capture complex dependencies between data, which has vastly improved the performance of downstream tasks [45,46] such as traffic flow prediction [47], recommender Systems [48], text classification [49], and action recognition [50]. GCNs [51] are multilayer graph convolutional networks that perform first-order local approximations to spectral graph convolutions neural networks, which solves the problem of its inability to maintain translation invariance on discrete non-Euclidean data while preserving the CNN's ability to process spatial features efficiently. As the third part of Figure 3 demonstrates, in the CSGNN framework, we select the GCN as the backbone of the GNN module, which is an essential component of the CSGNN framework. The sampled instance pairs are used to train the GCN-based contrastive learning model. We perform operations on each batch of instance pairs $I_i = (s_i, \mathcal{G}_i, y_i)$, where s_i denote the sampled samples in the instance pair, $\mathcal{G}_i$ denotes the group of adjacent samples in the instance pair, and y_i denotes the labels of the sampled samples. The GCN-based contrastive learning model mainly consists of a GCN module, a dimensionality reduction module, and an embedding comparison recognition module.

GCN module. This module efficiently mines the feature information of the sampled sample s_i and the adjacent sample group $\mathcal{G}_i$, mapping the embeddings of these two parts to the same embedding space, which will prepare the features for comparison between these two parts later. The layer-by-layer propagation principle of GCN for the adjacent sample group is shown in Equation (4).

$$\mathbf{H}_i^{(\ell+1)} = \phi\left(\widetilde{\mathbf{D}}_i^{-\frac{1}{2}} \widetilde{\mathbf{A}}_i \widetilde{\mathbf{D}}_i^{-\frac{1}{2}} \mathbf{H}_i^{(\ell)} \mathbf{W}^{(\ell)}\right) \tag{4}$$

Here, $\mathbf{H}_i^{(\ell)}$ refers to the representation matrix learned by the ℓ-th implicit layer, the input $\mathbf{H}_i^{(0)} = \mathbf{X}_i$, where $\mathbf{X}_i$ is the matrix of attribute vectors, and the output is labeled as the embedding $\mathbf{E}_i$ of the adjacent sample group $\mathcal{G}_i$. $\widetilde{\mathbf{D}}_i$ refers to the degree matrix of the adjacent sample group $\mathcal{G}_i$. $\widetilde{\mathbf{A}}_i = \mathbf{A}_i + \mathbf{I}$ refers to the adjacency matrix with self-connected subgraph, where $\mathbf{I}$ refers to the identity matrix. $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d^{(\ell)} \times d^{(\ell+1)}}$ refers to the ℓ-th layer trainable weight matrix. $\phi(\cdot)$ denotes the activation function such as ReLU.

Compared with the adjacent sample group $\mathcal{G}_i$, the sampled sample s_i has no structural information, so we only need to use the weight matrix of GCN and the corresponding activation function to complete its attribute information feature conversion, as shown in Equation (5).

$$\mathbf{z}_i^{(\ell+1)} = \sigma\left(\mathbf{z}_i^{(\ell)} \mathbf{W}^{(\ell)}\right) \tag{5}$$

where $\mathbf{z}_i^{(\ell)}$ is the row vector of feature representations of the sampled sample s_i learned by the ℓ-th implicit layer. $\mathbf{W}^{(\ell)}$ is the weight matrix shared with the GCN. The input $\mathbf{z}_i^{(0)}$ is defined as the row vector of attributes of the sampled sample s_i, and the output is labeled as the embedding $\mathbf{e}_i^{ss}$ of the sampled sample s_i.

Embedding comparison recognition module. This module maps the high-dimensional sample embedding $\mathbf{E}_i$ in the adjacent sample group $\mathcal{G}_i$ to the low-dimensional embedding space, facilitating comparison with the low-dimensional embedding feature $\mathbf{e}_i^{ss}$ of the sampled samples. The principle is shown in Equation (6).

$$\mathbf{e}_i^{as} = Dim^{reduction}(\mathbf{E}_i) = \sum_{m=1}^{n_i} \frac{(\mathbf{E}_i)_m}{n_i} \tag{6}$$

where $(\mathbf{E}_i)_m$ refers to the m-th rows of the adjacent sample group embedded in $\mathbf{E}_i$, and n_i refers to the sample size in the adjacent sample group $\mathcal{G}_i$.

Dimensionality reduction module. This module completes the embedding comparison of sampled samples and adjacent sample groups and is a vital part of the GNN-based contrastive learning model. Inspired by the literature [40], we apply a simple bilinear scoring function to this module. Equation (7) shows the details.

$$Comparator(\mathbf{e}_i^{as}, \mathbf{e}_i^{ss}) = \sigma\left(\mathbf{e}_i^{as} \mathbf{W}^{(b)} \mathbf{e}_i^{ssT}\right) \tag{7}$$

where $\mathbf{W}^{(b)}$ refers to the weight matrix of the comparison recognition module and $\phi(\cdot)$ denotes the sigmoid function.

Loss function. We use the standard binary cross-entropy (BCE) loss, applied to contrastive self-supervised learning tasks [20]. Unlike the BCE applied to category balancing as elaborated in the original paper, to effectively deal with the problem of category imbalance in food testing tasks, we perform balanced sampling between positive and negative instance pairs. Therefore, we use a common BCE. We follow Equation (8) to calculate separately the total The prediction score p_i for each instance pair $I_i = (s_i, \mathcal{G}_i, y_i)$ for each batch with batch number M. The loss function operation performed on I_i is shown in Equation (9).

$$p_i = CLM(s_i, \mathcal{G}_i) \tag{8}$$

$$\mathcal{L} = -\sum_{i=1}^{M} y_i \log(p_i) + (1 - y_i)\log(1 - p_i) \tag{9}$$

Here $CLM(\cdot)$ denotes the contrastive learning model.

3.2.4. Contamination Assessment

This part corresponds to the fourth part of Figure 3. After the GNN-based contrastive learning is completed, We assess the corresponding sample's hazardous contamination by identifying the consistency between the sampled sample s_i and the adjacent sample group $\mathcal{G}_i$. Ideally, the lower a sample's hazardous contamination, the closer the prediction score of its positive and negative instance pairs is to the median value (0.5). Conversely, the higher the hazardous contamination of a sample, the closer the prediction scores of its positive and negative instance pairs are to either side of 0 or 1. We define the contamination value of a sample as the absolute value of the difference between positive and negative instance pairs. Considering the incompleteness and chance of adjacent sample group $\mathcal{G}_i$ selection, we use multiple rounds of sampling to sample the testing samples. Specifically, we sample each sample in the detection data and sample the positive and negative instance pairs using the sampling strategy introduced in Section 3.2.2. The sampled instance pairs I_i are fed into the contrastive learning model, and their prediction scores p_i are calculated separately according to Equation (8).

Finally, the sample $f(s_i)$ value-at-contamination is calculated by averaging the absolute values of the prediction scores between positive and negative pairs of instances by subtraction over multiple rounds of sampling, as shown in Equation (10).

$$f(s_i) = \frac{\left|\sum_{r=1}^{R}\left(p_{i,r}^{(-)} - p_{i,r}^{(+)}\right)\right|}{R} \tag{10}$$

where R is the number of sampling rounds. $f(\cdot)$ is the mapping function of the detected data contamination values, which is the final objective function of the CSGNN framework.

3.3. *Evaluation Metrics*

The objective of the CSGNN framework in the contamination assessment process is to detect unqualified samples and complete the contamination hierarchy for qualified samples. We selected the following five evaluation metrics based on this objective.

The area under the ROC curve AUC will combine the ability of the model to detect qualified and unqualified samples and reasonably assess the comprehensive performance of the CSGNN framework in a dataset with category imbalance. Recall of unqualified samples, which reflects the probability that an unqualified sample is mistakenly detected as a qualified sample, is used to evaluate the model's ability to detect all unqualified samples. Precision reflects the probability that all detected unqualified samples are actually unqualified samples. We will use precision (all samples) and precision of qualified samples to measure the model's ability to identify unqualified samples and qualified samples, respectively. The false acceptance rate (FAR), which reflects the probability that the qualified sample is detected as an unqualified sample, is used to measure the model's contamination warning ability for qualified samples.

The meanings of the four base indicators TP, FP, FN, and TN in the confusion matrix are shown in Table 5. We mark the unqualified samples in the dataset as 1 and the qualified samples as 0. The evaluation metrics AUC, precision, the precision of qualified samples, recall of unqualified samples, and FAR are calculated in Equations (11)–(15), respectively.

$$AUC = \frac{\sum I(\text{ test qualified } > \text{ test unqualfied })}{(TP+FN)*(FP+TN)} \tag{11}$$

$$Precision = \frac{TP}{TP+FP} = \frac{\text{The number of unqualified samples correctly tested}}{\text{The number of samples tested as unqualified}} \tag{12}$$

$$Precision_{\text{qualified}} = \frac{TN}{TN+FN} = \frac{\text{The number of qualified samples correctly tested}}{\text{The number of samples tested as qualified}} \tag{13}$$

$$Recall_{\text{unqualified}} = \frac{TP}{TP+FN} = \frac{\text{The number of unqualified samples correctly tested}}{\text{The number of unqualified samples}} \tag{14}$$

$$FAR = \frac{FP}{FP+TN} = \frac{\text{The number of falsely tested as unqualified}}{\text{Total number qualified samples}} \tag{15}$$

Table 5. Meaning of base indicators in confusion matrix.

		Real Label	
		1	**0**
Predicted Label	1	True Positive (TP)	False Positive (FP)
	0	False Negative (FN)	True Negative (TN)

3.4. Baseline Model

We selected three supervised models and two unsupervised models as the baseline model. The supervised baseline model is chosen as NNLM, CNN, and GCN, and the unsupervised baseline model is chosen as LOF and GAN. It is worth noting that the baseline model uses the same data preprocessing as CSGNN and the three supervised models use the same dataset partitioning to ensure the fairness of the comparison.

3.4.1. NNLM

NNLM is a classic shallow neural network model in natural language processing, which proposes the introduction of word vectors in the model for the first time, and successfully breaks through the limitations of the N-gram model in modeling the relationship between words and words. NNLM is an excellent way to learn complex relationships between words and has played a role in the detection task [52], so it is used as the first baseline model in this paper. We set the number of hidden layer neurons to 16, the learning rate to 0.00001, the batch size to 16, and the epoch to 30.

3.4.2. CNN

CNN's expertise in capturing local feature information in data is widely used in image recognition and speech recognition. It has recently been applied to biometrics and food safety tasks with significant results [53]. We will use CNN as the second baseline model of this paper, setting two different convolutional kernels with four each, the activation function as ReLU, the optimizer as Adam, the learning rate as 0.001, the batch size as 32, and the epoch as 20.

3.4.3. GCN

GCN can simultaneously mine the attribute and structural information in the topology diagram for end-to-end learning, the mainstream GNN model. GCN has more robust feature extraction capabilities than CNNs, solving the problem that CNNs cannot maintain panning invariant in non-Euclidean data, and can effectively mine complex correlation information in data, showing promising potential in classification tasks [54]. Therefore, we will use GCN as the third baseline model in this paper to explore the effect of the GNN algorithm in the food detection task, setting up two convolutional layers, the activation function as ReLU, the optimizer as Adam, with the learning rate of 0.01 and the epoch of 200.

3.4.4. LOF

LOF is a density-based unsupervised anomaly detection algorithm that determines whether a data point is abnormal by comparing the local neighbor density of each data point with its neighborhood data point [55]. Inspired by anomaly detection in attribute networks, we found that anomaly detection has a similar principle to food detection tasks, with the goal of anomalous mining data with feature information that differs from most of the data. Therefore, we will use LOF as the fourth baseline model for this paper to explore the effect of this class of anomaly detection algorithms in food detection tasks.

3.4.5. GAN

GAN comprises the generator (G) and discriminator (D), a generative model based on unsupervised learning methods widely used in image generation and style migration. It also exhibited good performance in the detection and classification tasks [56], so this paper used GAN as the fifth baseline model. We chose Adam as the optimizer for G and D, ReLU as the activation function, the learning rate is 0.0001, the batch size is 32, and the epoch is 500.

3.5. Parameter Settings

In the structured representation phase of the data, we construct network $\mathcal{G}$ in the same way as the GCN model in the baseline model. During the contrastive instance pair sampling phase, we set the size of the instance pair to $\mathcal{G}_i$ of the adjacent sample group in the I_i to 5. The available nodes will be reused to reach the set size for nodes smaller than the adjacent sample group $\mathcal{G}_i$ fixed size. In the GCN-based contrastive learning phase, set the number of module layers to 1 and the embedded dimension fixed to 6. We chose Adam as the optimizer. In addition, Set the learning rate to 0.006, the batch size to 450, and the epoch to 1000. We set the sampling round R to 256.

4. Experiments and Analysis of Results

In this section, we conducted a complete experiment and gave a detailed experimental comparison and analysis. We explain the entire experimental validity by the following three main questions.

Q1: What are the advantages of the CSGNN framework over the baseline model? What performance will these advantages demonstrate in real-world application scenarios?

Q2: What kind of contamination warning does the CSGNN framework enable in contamination assessment applications for food quality? How does it do it?

Q3: Is there a reasonable and feasible explanation for the contamination classification of the CSGNN framework in contamination assessment for food quality?

We will answer each of these questions in the following content and elaborate on the details of our experiment.

4.1. Analysis of Results

4.1.1. For Q1 (What Are the Advantages of the CSGNN Framework over the Baseline Model? What Performance Will These Advantages Demonstrate in Real-World Application Scenarios?)

We completed a comparison experiment between five baselines and CSGNN models on the sterilized dairy product detection data, as shown in Table 6. Overall, the CSGNN model performed better than all baseline models. Specifically, we have the following findings.

Table 6. All models were evaluated 30 times on the sterilized milk detection dataset and averaged the results. (In input, X indicates the input data information, and Y represents the Label corresponding to the data. The best performance values of each evaluation metric in the supervised and unsupervised models are shown in bold and bold combined with underlining, respectively. * indicates the best performing model applicable to the food contamination early warning task in each evaluation metric).

Input	Models	AUC	Precision	Precision of Qualified Samples	Recall of Unqualified Samples	FAR
(X,Y)	NNLM	0.7602	0.9668	0.9904	0.7059	**0.2941 ***
	CNN	0.6765	0.9833	0.9829	0.3529	0.6470
	GCN	**0.9988**	**0.9979**	**1.0000**	**1.0000**	0.0024
(X)	LOF	**0.9150**	0.9787	0.9787	0.8823	0.0523
	GAN	0.5804	0.9546	0.9782	0.2353	0.7647
	CSGNN	0.9140	**0.9829**	**1.0000**	**1.0000**	**0.1719 ***

Firstly, in all models, the AUC values of GCN, LOF, and CSGNN were higher than 0.91. The AUC value of GCN is the highest value of 0.9988 of all models, indicating that the GNN algorithm exhibits very stable performance in food detection tasks. For the AUC values in unsupervised models, the LOF model performed best at 0.9150, possibly because the goal of the anomaly detection task was to find a small number of anomalies in most of the data, and its purpose was to target categorical imbalances. The CSGNN is second only to the LOF model 0.001, indicating that the GCN-based contrastive learning module in the CSGNN framework can show stable performance when dealing with categorical imbalance data, which is critical in practical application scenarios.

Secondly, all models have precision values above 0.95. However, the precision values for GCN and CSGNN are the highest in supervised and unsupervised models. For supervised models, the precision value of GCN 0.9979 is 0.0146 higher than that of the sub-optimal model CNN. For the unsupervised model, the precision value of CSGNN is 0.9829 higher than that of the sub-optimal model LOF. In addition, both GCN and CSGNN had a precision of qualified sample value of 1.0000, which was 0.0096 more heightened than the sub-optimal model NNLM in all models. This shows that the GNN algorithm can better identify qualified and unqualified samples by mining complex correlation information in the detection data, showing promising potential in food detection tasks.

Thirdly, the recall of the unqualified sample value of both GCN and CSGNN is 1.0000, which is more than 13% higher than the sub-optimal model LOF in all models. The recall of unqualified samples reflects the model's ability to check the unqualified samples in the detection data, which is the essential task for contamination assessment of food quality. For other baseline models, the reason they cannot successfully detect all the unqualified samples may be a bottleneck in mining complex feature information between samples of detection data. The two GNN models can accurately detect all unqualified samples, indicating that the GNN algorithm has successfully captured the detection data's attribute information and topology information in the food detection task and has good application potential. Further, the CSGNN framework provides a solution for GNN algorithms in unsupervised applications.

Fourthly, the contamination warning task for food quality aims to mine unqualified and potentially contaminated qualified samples in the detection data. To smoothly identify potentially contaminated qualified samples, the FAR value in the food detection task is not as low as possible because the FAR value that is too low will not be conducive to our contamination classification of qualified samples. For the FAR value of the supervised model, NNLM is more suitable for the contamination classification task of detection data. The performance of GCN and CNN is too low and high, respectively, which is not conducive to good contamination classification. For the FAR values of the unsupervised model, LOF and GAN are too low and too high, respectively, and are unsuitable for contamination classification. CSGNN can better achieve the contamination classification of detection data because the CSGNN framework effectively solves the problem that GNN is too low in this metric by setting hyperparameters. The specific scheme is detailed in Section 4.1.2.

4.1.2. For Q2 (What Kind of Contamination Warning Does the CSGNN Framework Enable in Contamination Assessment Applications for Food Quality? How Does It Do It?)

Specifically, the CSGNN framework is divided into three stages: preprocessing and composition, training learning, and contamination assessment. In the preprocessing and composition stage, the unitless processing of the detection data is realized, and the structured representation is completed. During the training learning stage, the contrastive learning model completes the training of the instance pair in a self-supervised manner. In the contamination assessment stage, the contamination value of each sample is obtained, and both the detection data's binomial classification and contamination level classification is completed accordingly.

We define the contamination value of each sample as the absolute value of its prediction score subtracted between positive and negative instance pairs. In the experiment, we labeled the lowest value of the contamination value of the unqualified samples in the dataset as W. W, to some extent, reflects the boundary value of the contamination value of the samples with higher contamination probability in the dataset, which is the reason why we set it as the threshold value in the process of dichotomous determination between qualified and unqualified samples by the model for the detection data. Ideally, the greater the hazardous contamination of the contaminated samples after pretreatment, the more the prediction scores of positive and negative instance pairs are distributed to both sides of 0 and 1. The prediction scores of positive and negative instance pairs are close to the qualified samples' median value (0.5). Accordingly, the contamination value of U = 0.5 can be used as a conservative boundary value to distinguish the contaminated samples from the negligible contamination samples by default. We classify the testing samples into four contamination classes based on the contamination value: negligible, low, medium, and high. Each contamination level is classified based on the following.

- Negligible contamination: negligible contamination level, samples are basically risk-free. $0 \leqslant$ contamination value $<U$ for qualified samples.
- Low contamination: Low contamination level, samples with a low probability of being at risk. $U \leqslant$ contamination value $<W$ for qualified samples.
- Medium contamination: Medium contamination level, samples with a high probability of being at risk. $W \leqslant$ contamination value $\leqslant 1$ for qualified samples.
- High contamination: High contamination level, all unqualified samples belong to this category.

To observe the contamination warning effect of the CSGNN framework more intuitively, we made a visual display, as shown in Figure 7. Based on the distribution of contamination values of the samples, they are sequentially classified as negligible contamination samples, low contamination samples, medium contamination samples, and high contamination samples.

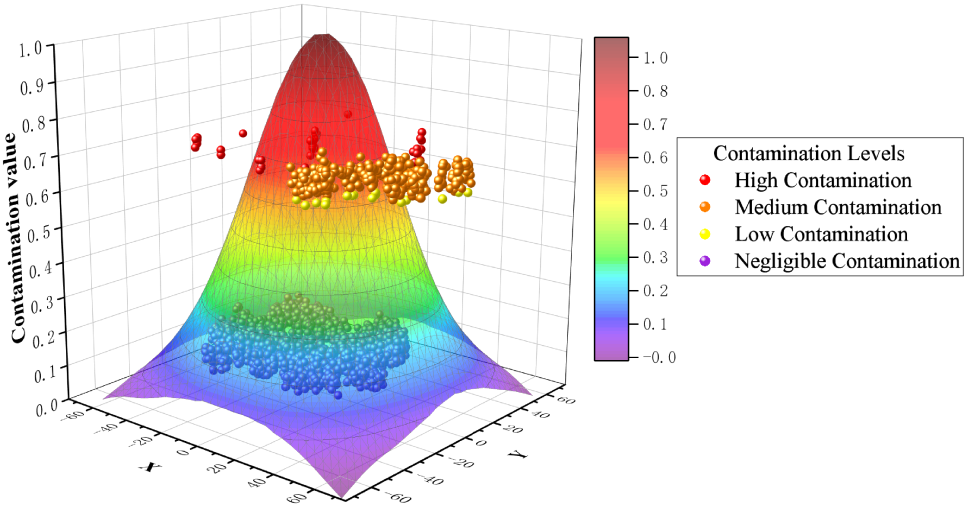


Figure 7. Contamination ranking of the CSGNN framework on sterilized dairy products detection dataset.

4.1.3. For Q3 (Is There a Reasonable and Feasible Explanation for the Contamination Classification of the CSGNN Framework in Contamination Assessment for Food Quality?)

From Figure 7, it can be found that the same contamination level samples exist in clusters in the graph with a minimal distance between samples. This is because the difference between the contamination values of samples with the same contamination level is minimal. At the same time, there is a significant difference between the contamination values of samples with different contamination levels. No sample does not comply with the principle, which indicates that our principle of dividing samples with varying levels of contamination is feasible. Specifically, for the 2158 sterilized milk detection data, 1734 of the 2117 qualified samples had contamination values below 0.17 and were classified as negligible contamination by the CSGNN framework. 41 unqualified samples had a minimum contamination value of $W = 0.674$ and were all classified as high contamination. 19 qualified samples had contamination values above 0.17, ranked in the low contamination sample class. The 364 qualified samples with contamination values between W and 1 were classified as medium contamination samples. As shown in Figure 8, The CSGNN framework successfully detects all unqualified samples from the original sterilized milk detection data while classifying the qualified samples specifically classified into three levels according to the contamination value: medium contamination, low contamination and negligible contamination. To further observe the detection performance of the CSGNN, As shown in Figure 9, we checked the contamination values of the three nearest unqualified samples A, B, and C, which was 0.6744, 0.6782, and 0.6789, respectively, for W. We further examined the specific values of the six evaluation metrics for these three samples, where sample A unqualified due to its fat content of 3.57, which was below the standard range minimum value of 3.7, and sample B was unqualified because its nonfat milk solids content was 4.69, which was lower than the minimum value of 8.5 in the standard range. Sample C was unqualified because its acidity content was 10.90, which did not meet the standard range of 11$\sim$16.

In addition, for the contamination values of negligible contamination and contaminated samples in the dataset, the CSGNN shows a significant difference (A cliff-like gap between the contamination values of negligible contamination and contaminated samples), and the boundary value distinguishing the two is much lower than the default U value, even only 0.17. our framework highlights the information difference between the negligible contamination and contaminated samples in the preprocessing process. The chance judgment caused by local perception is smoothly avoided in the contrastive learning process of multiple rounds of sampled instance pairs. The prediction scores of the instance pairs corresponding to the negligible contamination samples are realized to be infinitely close to the median value (0.5) in calculating the contamination value. In contrast, the prediction scores of the instance pairs corresponding to the contaminated samples are distributed to both sides of 0 and 1. In summary, the CSGNN framework has a significant hierarchy of contamination values for sterilized dairy product detection samples and efficiently achieves a reasonable classification of food safety contamination levels.

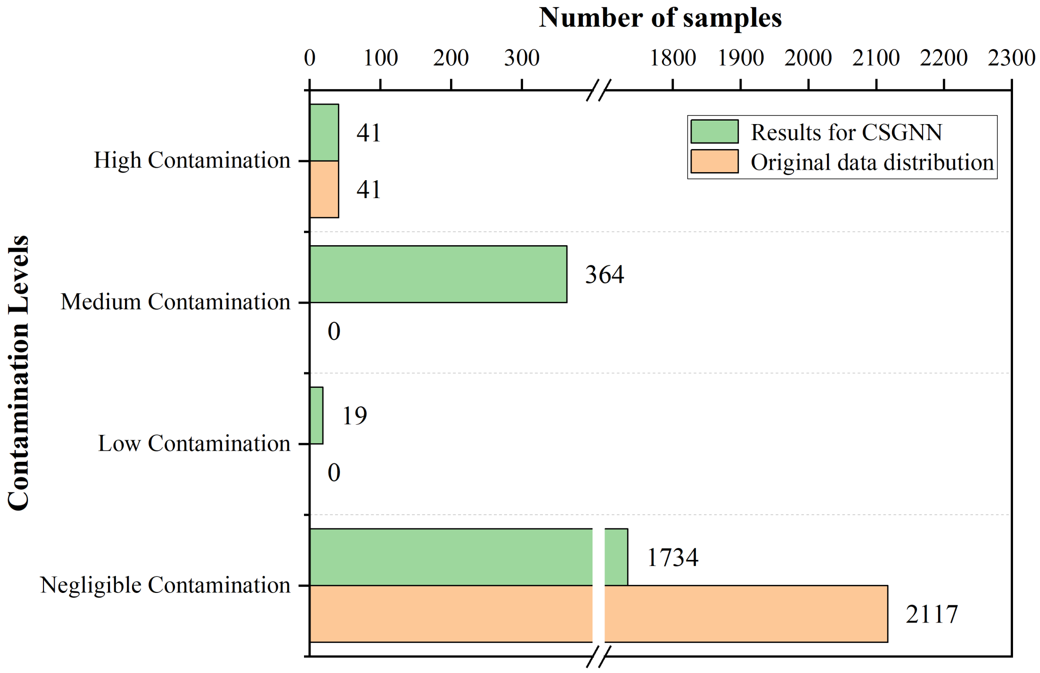


Figure 8. Division of sample numbers for each contamination level of the CSGNN framework on the sterilized dairy products detection dataset.

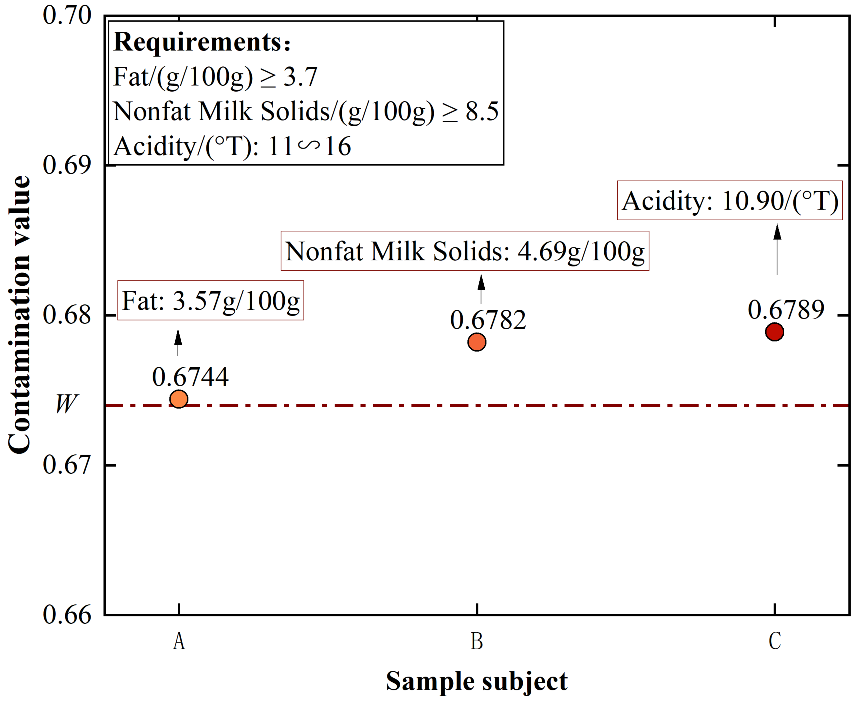


Figure 9. The specifics of three unqualified sample objects whose contamination values from the CSGNN framework are closest to the *W* values.

The CSGNN framework performs data preprocessing and optimal composition for unknown food detection data. The data are fed into a feedforward neural network to obtain the contamination value of each testing sample. This process focuses on the lowest contamination value W of the unqualified samples and the more obvious boundary value U between the contaminated and negligible contamination samples. (Considering each detection data's variable quality, we will default U = 0.5 if no obvious U value is observed). The framework will achieve the contamination classification of different food detection data based on the W and U values obtained from different data according to the principles described in Section 4.1.2.

4.2. Application and Optimization of the CSGNN

Inspired by [57] and considering the rigor of government work, we introduce a panel of experts from food quality supervision departments to participate in the example analysis and application. The CSGNN framework can provide technical support and the theoretical basis for implementing food safety supervision and effectively improve the efficiency of food quality contamination warnings by the expert team in the detection of massive, diverse types and complex relationships of food data in practical application scenarios. The expert team analyzes and implements more refined regulatory decisions based on CSGNN's quality contamination warning results, ensuring the reasonableness and stability of the safety and quality assessment results.

Our intelligent food safety supervision platform, built in cooperation with the Institute of Product Quality Supervision and Inspection in a Chinese province, promotes the application of CSGNN on the ground. We encapsulated the CSGNN framework to provide technical support for this platform's hazard contamination warning module. For a batch of detection data of any food type, CSGNN takes the detection data of N samples V detection indicators $x_{v1,\ldots,}x_{vn}$ as input (as shown in the "Item" column in Table 3), and the output is the contamination level of N samples. Our framework shows good detection performance during online testing of the input data, and feeds the results to the terminal according to the contamination level. The regulatory staff analyzes the results and takes appropriate warning measures to reduce the adverse effects of hazardous contaminated products.

5. Discussion

CSGNN provides an efficient end-to-end approach for contamination warning and control of food quality based on self-supervised learning, which can generate significant social and economic benefits while consuming lower costs. Specifically, the CSGNN model can quickly perform food quality detection on an ordinary computer, requiring minimal financial and time costs and no human intervention. At the same time, the model can effectively optimize the food safety and quality control system in an early warning way, and reduce the economic loss (financial penalty) and sales loss due to the damage to the company's reputation caused by unqualified products. For example, a food safety incident occurred in 2022 in which propylene glycol was detected in pure milk produced by Maiquer Group (https://www.samr.gov.cn/xw/zj/202207/t20220703_348326.html accessed date: 15 February 2023), which not only caused the company to be fined tens of millions of dollars (https://static.cninfo.com.cn/finalpage/2022-08-23/1214356756.PDF accessed date: 15 February 2023). More importantly, it seriously affected the brand's trust in consumers' minds and caused irreparable cascading negative impacts on the company's business performance.

Combined with the analysis of the results of the contamination classification provided by CSGNN, food safety regulators will establish an emergency mechanism to deal with special hazards. Specifically, recall of unqualified samples from food detection data and further traceability of testing indicators that lead to samples with hazard contamination. Thus, a priority system for hazard analysis and effective measures for safety regulation will be established, and unified decisions on profit and loss of economic and food safety will be taken from a global perspective. Hazard contamination of food is closely related to food producers, and the theoretical basis provided by the CSGNN detection results will help regulators to establish food safety control strategies based on scientific principles and strengthen the effectiveness of regulation of food producers. Thus, they are urged to focus on the content of testing indicators that often lead to samples with hazardous contamination in the production process. This can optimize the preventive principle of hazard contamination in food production and processing, and efficiently supervise and guide food producers' benign production to minimize the occurrence of food hazard contamination accidents.

6. Conclusions and Future Work

This paper proposes applying GNN-based contrastive self-supervised learning to contamination warning and control of food quality. This is the first attempt at GNN-based self-supervised learning in food safety early warning analysis. We innovatively proposed an end-to-end contamination assessment framework for food quality called CSGNN. The framework directly models the complex feature associations between the detection data. It uses contrastive self-supervised learning to construct positive and negative instance pairs to train the detection data of category imbalance. Finally realizes the detection of unqualified samples and the contamination level of qualified samples by obtaining the contamination value of each detection sample. We applied the framework to a batch of sterilized milk detection data in a province in China, and its recall of unqualified samples and AUC values reached 1.0000 and 0.9188, respectively, which indicates that our framework can detect all unqualified samples and show better stability and lower false detection rate in the practical application of data category imbalance. Experimental results show that the CSGNN framework successfully mines the attribute information and structural information between different indicators of food detection data, and the final contamination value of the detection sample reflects its contamination level well to achieve efficiently. Our research provides new ideas for contamination assessment of food quality, and food safety regulators can make more efficient decisions based on CSGNN detection results combined with expert panels.

There is a degree of subjective interference in the parameters of CSGNN in contamination classification (which is set to 0.5 by default when a more conservative boundary value U between contaminated and negligible contamination samples is not observable in the detection data). In future work, we will further explore using a parameter-free approach to risk ranking to ensure the objectivity of the process, e.g., data analysis of contamination values of samples utilizing mathematical statistics to determine the contamination rank of each sample. In addition, we will more fully consider information on food hazard contamination and toxicology to optimize the contamination assessment model for food quality in subsequent studies.

Author Contributions: Conceptualization, J.Y. and H.L.; methodology, J.Y. and E.Z; validation, T.L. and C.C. (Chen Chen); formal analysis, C.C. (Cheng Chen); investigation, T.L.; data curation, H.L.; writing—original draft preparation, J.Y.; writing—review and editing, E.Z.; visualization, X.L.; supervision, E.Z. and X.L.; project administration, X.L.; funding acquisition, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China (2019YFC1606100 and sub-program 2019YFC1606104), the Major science and technology projects of Xinjiang Uygur Autonomous Region (2020A03001, sub-program 2020A03001-1 and 2020A03001-3), and the special scientific research project for young medical science (2019Q003).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Food and Agriculture Organization. Maximum Residue Limits (mrls) and Risk Management Recommendations (rmrs) for Residues of Veterinary Drugs in Foods-cx/mrl 2-2018. 2018. Available online: https://dokumen.tips/documents/maximum-residuelimitsmrlsandriskmanagement.html (accessed on 12 August 2022).
2. Health Canada. List of Maximum Residue Limits (mrls) for Veterinary Drugs in Foods. 2022. Available online: https://www.canada.ca/en/health-canada/services/drugs-health-products/veterinary-drugs/maximum-residue-limits-mrls/list-maximum-residue-limits-mrls-veterinary-drugs-foods.html (accessed on 1 September 2022).
3. Mensah, L.D.; Julien, D. Implementation of food safety management systems in the uk. *Food Control* **2011**, 22, 1216–1225. [CrossRef]
4. Neltner, T.G.; Kulkarni, N.R.; Alger, H.M.; Maffini, M.V.; Bongard, E.D.; Fortin, N.D.; Olson, E.D. Navigating the us food additive regulatory program. *Compr. Rev. Food Sci. Food Saf.* **2011**, 10, 342–368. [CrossRef]
5. Jen, J.J.-S.; Chen, J. *Food Safety in China: Science, Technology, Management and Regulation*; John Wiley & Sons: Hoboken, NJ, USA, 2017.
6. Wu, Y.-N.; Liu, P.; Chen, J.-S. Food safety risk assessment in china: Past, present and future. *Food Control* **2018**, *90*, 212–221. [CrossRef]
7. Han, Y.; Cui, S.; Geng, Z.; Chu, C.; Chen, K.; Wang, Y. Food quality and safety risk assessment using a novel hmm method based on gra. *Food Control* **2019**, *105*, 180–189. [CrossRef]
8. Geng, Z.; Shang, D.; Han, Y.; Zhong, Y. Early warning modeling and analysis based on a deep radial basis function neural network integrating an analytic hierarchy process: A case study for food safety. *Food Control* **2019**, *96*, 329–342. [CrossRef]
9. Geng, Z.; Liu, F.; Shang, D.; Han, Y.; Shang, Y.; Chu, C. Early warning and control of food safety risk using an improved ahc-rbf neural network integrating ahp-ew. *J. Food Eng.* **2021**, *292*, 110239. [CrossRef]
10. Smid, J.H.; Verloo, D.; Barker, G.C.; Havelaar, A.H. Strengths and weaknesses of monte carlo simulation models and bayesian belief networks in microbial risk assessment. *Int. J. Food Microbiol.* **2010**, *139*, S57–S63. [CrossRef]
11. Soon, J.M. Application of bayesian network modelling to predict food fraud products from china. *Food Control* **2020**, *114*, 107232. [CrossRef]
12. Geng, Z.Q.; Zhao, S.S.; Tao, G.C.; Han, Y.M. Early warning modeling and analysis based on analytic hierarchy process integrated extreme learning machine (ahp-elm): Application to food safety. *Food Control* **2017**, *78*, 33–42. [CrossRef]
13. Goldberg, D.M.; Khan, S.; Zaman, N.; Gruss, R.J.; Abrahams, A.S. Text mining approaches for postmarket food safety surveillance using online media. *Risk Anal.* **2020**. [CrossRef]
14. Weller, D.L.; Love, T.M.T.; Wiedmann, M. Comparison of resampling algorithms to address class imbalance when developing machine learning models to predict foodborne pathogen presence in agricultural water. front. *Environ. Sci* **2021**, *9*, 701288. [CrossRef]
15. Seiffert, C.; Khoshgoftaar, T.M.; Hulse, J.V.; Napolitano, A. Rusboost: A hybrid approach to alleviating class imbalance. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2009**, *40*, 185–197. [CrossRef]
16. Sun, Y.; Kamel, M.S.; Wong, A.K.C.; Wang, Y. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognit.* **2007**, *40*, 3358–3378. [CrossRef]
17. Pöppelbaum, J.; Chadha, G.S.; Schwung, A. Contrastive learning based self-supervised time-series analysis. *Appl. Soft Comput.* **2022**, *117*, 108397. [CrossRef]
18. Liu, Y.; Pan, S.; Jin, M.; Zhou, C.; Xia, F.; Yu, P.S. Graph self-supervised learning: A survey. *arXiv* **2021**, arXiv:2103.00111.
19. Velickovic, P.; Fedus, W.; Hamilton, W.L.; Liò, P.; Bengio, Y.; Hjelm, R.D. Deep graph infomax. *ICLR (Poster)* **2019**, *2*, 4.
20. Hassani, K.; Khasahmadi, A.H. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*; PMLR: New York, NY, USA, 2020; pp. 4116–4126.
21. Liu, Y.; Li, Z.; Pan, S.; Gong, C.; Zhou, C.; Karypis, G. Anomaly detection on attributed networks via contrastive self-supervised learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 2378–2392. [CrossRef]
22. Moayedikia, A. Multi-objective community detection algorithm with node importance analysis in attributed networks. *Appl. Soft Comput.* **2018**, *67*, 434–451. [CrossRef]
23. Li, Z.; Wang, X.; Li, J.; Zhang, Q. Deep attributed network representation learning of complex coupling and interaction. *Knowl.-Based Syst.* **2021**, *212*, 106618. [CrossRef]
24. Zhao, Z.; Zhou, H.; Li, C.; Tang, J.; Zeng, Q. Deepemlan: Deep embedding learning for attributed networks. *Inf. Sci.* **2021**, *543*, 382–397. [CrossRef]
25. Zuo, E.; Du, X.; Aysa, A.; Lv, X.; Muhammat, M.; Zhao, Y.; Ubul, K. Anomaly score-based risk early warning system for rapidly controlling food safety risk. *Foods* **2022**, *11*, 2076. [CrossRef] [PubMed]
26. Liao, L.; He, X.; Zhang, H.; Chua, T.-S. Attributed social network embedding. *IEEE Trans. Knowl. Data Eng.* **2018**, *30*, 2257–2270. [CrossRef]
27. Pan, S.; Hu, R.; Fung, S.-F.; Long, G.; Jiang, J.; Zhang, C. Learning graph embedding with adversarial training methods. *IEEE Trans. Cybern.* **2019**, *50*, 2475–2487. [CrossRef]
28. Wang, X.; Bouzembrak, Y.; Lansink, A.G.J.M.O.; van der Fels-Klerx, H.J. Application of machine learning to the monitoring and prediction of food safety: A review. *Compr. Rev. Food Sci. Food Saf.* **2022**, *21*, 416–434. [CrossRef] [PubMed]
29. Widodo, A.; Yang, B.-S. Support vector machine in machine condition monitoring and fault diagnosis. *Mech. Syst. Signal Process.* **2007**, *21*, 2560–2574. [CrossRef]

30. Bornn, L.; Farrar, C.R.; Park, G.; Farinholt, K. Structural health monitoring with autoregressive support vector machines. *J. Vib. Acoust.* **2009**, *131*. [CrossRef]
31. Yang, Y.; Wei, L.; Pei, J. Application of bayesian modelling to assess food quality & safety status and identify risky food in china market. *Food Control* **2019**, *100*, 111–116.
32. Wei, L.; Pei, J.; Zhang, A.; Wu, X.; Xie, Y.; Yang, Y. Application of stochastic bayesian modeling to assess safety status of baby formulas and quantify factors leading to unsafe products in china market. *Food Control* **2020**, *108*, 106826. [CrossRef]
33. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
34. Coudray, N.; Ocampo, P.S.; Sakellaropoulos, T.; Narula, N.; Snuderl, M.; Fenyö, D.; Moreira, A.L.; Razavian, N.; Tsirigos, A. Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. *Nat. Med.* **2018**, *24*, 1559–1567. [CrossRef]
35. Nogales, A.; Díaz-Morón, R.; García-Tejedor, Á.J. A comparison of neural and non-neural machine learning models for food safety risk prediction with european union rasff data. *Food Control* **2022**, *134*, 108697. [CrossRef]
36. Parisot, S.; Ktena, S.I.; Ferrante, E.; Lee, M.; Guerrero, R.; Glocker, B.; Rueckert, D. Disease prediction using graph convolutional networks: Application to autism spectrum disorder and alzheimer's disease. *Med. Image Anal.* **2018**, *48*, 117–130. [CrossRef] [PubMed]
37. Liu, X.; Zhang, F.; Hou, Z.; Mian, L.; Wang, Z.; Zhang, J.; Tang, J. Self-supervised learning: Generative or contrastive. *IEEE Trans. Knowl. Data Eng.* **2021**, *35*, 857–876. [CrossRef]
38. Hafidi, H.; Ghogho, M.; Ciblat, P.; Swami, A. Negative sampling strategies for contrastive self-supervised learning of graph representations. *Signal Process.* **2022**, *190*, 108310. [CrossRef]
39. Liu, Y.; Zheng, Y.; Zhang, D.; Chen, H.; Peng, H.; Pan, S. Towards unsupervised deep graph structure learning. *arXiv* **2022**, arXiv:2201.06367.
40. Van den Oord, A.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
41. Chen, X.; He, K. Exploring simple siamese representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 15750–15758.
42. Jintao, H. Food safety law of the people's republic of china. *Chin. L. Gov't* **2012**, *45*, 10.
43. Qiu, J.; Chen, Q.; Dong, Y.; Zhang, J.; Yang, H.; Ding, M.; Wang, K.; Tang, J. Gcc: Graph contrastive coding for graph neural network pre-training. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Long Beach, CA, USA, 6–10 July 2020; pp. 1150–1160.
44. Tong, H.; Faloutsos, C.; Pan, J.Y. Fast random walk with restart and its applications. In Proceedings of the Sixth International Conference on Data Mining (ICDM'06), Hong Kong, China, 18–22 December 2006; pp. 613–622.
45. Saha, P.; Mukherjee, D.; Singh, P.K.; Ahmadian, A.; Ferrara, M.; Sarkar, R. Retracted article: Graphcovidnet: A graph neural network based model for detecting COVID-19 from ct scans and x-rays of chest. *Sci. Rep.* **2021**, *11*, 8304. [CrossRef]
46. Jiang, P.; Huang, S.; Fu, Z.; Sun, Z.; Lakowski, T.M.; Hu, P. Deep graph embedding for prioritizing synergistic anticancer drug combinations. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 427–438. [CrossRef]
47. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3848–3858. [CrossRef]
48. Monti, F.; Bronstein, M.; Bresson, X. Geometric matrix completion with recurrent multi-graph neural networks. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. [CrossRef]
49. Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7370–7377.
50. Shi, L.; Zhang, Y.; Cheng, J.; Lu, H. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12026–12035.
51. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
52. Darliansyah, A.; Naeem, M.A.; Mirza, F.; Pears, R. Sentipede: A smart system for sentiment-based personality detection from short texts. *J. Univ. Comput. Sci.* **2019**, *25*,1323–1352.
53. Vo, S.A.; Scanlan, J.; Turner, P. An application of convolutional neural network to lobster grading in the southern rock lobster supply chain. *Food Control* **2020**, *113*, 107184. [CrossRef]
54. Hong, D.; Gao, L.; Yao, J.; Zhang, B.; Plaza, A.; Chanussot, J. Graph convolutional networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 5966–5978. [CrossRef]
55. Breunig, M.M.; Kriegel, H.; Ng, R.T.; Sander, J. Lof: Identifying density-based local outliers. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, TX, USA, 16–18 May 2000; pp. 93–104.

56. Kim, J.-Y.; Bu, S.-J.; Cho, S.-B. Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders. *Inf. Sci.* **2018**, *460*, 83–102. [CrossRef]
57. Zuo, E.; Aysa, A.; Muhammat, M.; Zhao, Y.; Chen, B.; Ubul, K. A food safety prescreening method with domain-specific information using online reviews. *J. Consum. Prot. Food Saf.* **2022**, *17*, 163–175. [CrossRef]

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