electronics

Article

Training Data Selection by Categorical Variables for Better Rare Event Prediction in Multiple Products Production Line

Dongting Xu [1,2], Zhisheng Zhang [1] and Jinfei Shi [1,2,*]

[1] School of Mechanical Engineering, Southeast University, Nanjing 211189, China; xudongting@seu.edu.cn (D.X.); oldbc@seu.edu.cn (Z.Z.)
[2] School of Mechanical Engineering, Nanjing Institute of Technology, Nanjing 211167, China
* Correspondence: shijf_xdt@163.com; Tel.: +86-02586118812

Abstract: Manufacturers are struggling to use data from multiple products production lines to predict rare events. Improving the quality of training data is a common way to improve the performance of algorithms. However, there is little research about how to select training data quantitatively. In this study, a training data selection method is proposed to improve the performance of deep learning models. The proposed method can represent different time length multivariate time series spilt by categorical variables and measure the (dis)similarities by the distance matrix and clustering method. The contributions are: (1) The proposed method can find the changes to the training data caused by categorical variables in a multivariate time series dataset; (2) according to the proposed method, the multivariate time series data from the production line can be clustered into many small training datasets; and (3) same structure but different parameters prediction models are built instead of one model which is different from the traditional way. In practice, the proposed method is applied in a real multiple products production line dataset and the result shows it can not only significantly improve the performance of the reconstruction model but it can also quantitively measure the (dis)similarities of the production behaviors.

Keywords: multivariate time series; categorical variables; Euclidian distance matrix; integrated feature representation; autoencoder

Citation: Xu, D.; Zhang, Z.; Shi, J. Training Data Selection by Categorical Variables for Better Rare Event Prediction in Multiple Products Production Line. *Electronics* **2022**, *11*, 1056. https://doi.org/10.3390/electronics11071056

Academic Editors: Katarzyna Antosz, Jose Machado, Yi Ren, Rochdi El Abdi, Dariusz Mazurkiewicz, Marina Ranga, Pierluigi Rea, Vijaya Kumar Manupati, Emilia Villani and Erika Ottaviano

Received: 28 February 2022
Accepted: 24 March 2022
Published: 28 March 2022

1. Introduction

Manufacturers are grabbing big data to transform to advanced digital manufacturing using statistical models or artificial intelligence (AI) technology. Machine learning or deep learning is tuned and tweaked to near-perfect performance in the lab but often fails to solve actual problems in real factory floor settings. The way we train AI is fundamentally flawed [1]. This phenomenon requires new ideas and technologies to support the less "confident" AI models to turn them into "more responsible AI". The performance of an AI algorithm relies on the quality of data, but there is little research about how to choose the right training data quantitatively. Prioritized efforts are needed to ensure proper data quality over developing the machine learning algorithms [2]. Increased awareness of data and pre-selecting good quality training data can help understand more about the production and reduce the risk of lack understanding the "black box". One big challenge is figuring out what data are important as well as measuring the quality of data quantitatively. To achieve this goal, domain knowledge from field experts and new technology are required to measure how efficient and effective the training data input are.

The new data streams coming from different directions provided by the Internet of Things (loT) make the problem more challenging. Today, manufacturers are getting smarter sensors located in the machines generating multivariate time series data which are the common form of data streams. Multivariate time series not only have a time-based dimension but also a variable-based dimension. The variables can be continuous variables

or categorical variables. However, difficulties are often caused by categorical variables since they are not easily used to build a model. During the process of building a machine learning model, a simple way is dropping the categorical variables which may cause insufficient use of data [3,4].

Deep learning models are used for rare event prediction in the paper manufacturing process [3]. The deep learning method is a one-stage method and it can perform well in failure prediction. Feature extraction methods and support vector machine which are supervised learning methods are used to predict the failures in a production line. The features are median, standard deviation, and the relative distance between the abnormal data and normal data [4]. To do feature engineering before training machine learning is a two-stage method. However, they both drop all the categorical variables and keep all the continuous variables to fit and train the model.

On a factory floor, since the failure data size is smaller compared to the size of the "normal" dataset, the dataset is always imbalanced. For many applications requiring supervised learning or semi-supervised learning, the key to fast and high-quality learning is a well-balanced training dataset. If the imbalanced datasets are fed into the supervised learning process, the algorithm often cannot provide satisfactory results. For imbalanced datasets, we often rely on simulation models or data augmentation to get more failure data to overcome the lack of positive data for supervised learning. The effective approaches of resampling techniques [5,6] can work well on univariate time series; however, for multivariate time series, there are few effective methods to generate more data which do not bring big changes to the original data [3]. In this situation, carefully selecting the training data is a way to improve the performance of machine learning algorithms.

With machine learning systems, the accuracy of real-world results from the model is highly dependent upon the quality of the selected training data [7]. The research direction of the training data selection algorithm and methods to improve data quality for machine learning will significantly change in the very near future [8,9]. The better the understanding of the training data, the better model we can choose to fit and test. Due to the challenges in obtaining labels, supervised learning algorithms are less attractive for the task of anomaly detection. Semi-supervised learning or unsupervised learning is becoming more popular. The reconstruction model reduces the need for supervision. Autoencoder is an effective deep learning model which can be used as a semi-supervised learning model as well as a reconstruction model [10]. The reconstruction error is the element used to detect anomalies or failures. If we use the reconstruction error method, the quality of the training data is important.

Engineers from General Electric Company (GE) described the paper making process as a multistage process [11]. For building the break indicator, they did data reduction, variable selection, value transformation, and model generation. According to their expert knowledge, some variables were selected but some were dropped to train the supervised learning models. However, they did not remove paper grade variability. Their study shows paper grade changes could cause significant changes in process variables. Bissessur et al. did some work on monitoring the performance of the paper making process [12]. They built a vibration-based condition monitoring system. The key data are the vibration signals. The neural network classifier was built based on the vibration. Their work did not sufficiently use information about paper grade or other signals from the paper machines.

The similarity of time series is often measured by distance [13,14]. A covariance matrix is widely used for developing feature extraction methods. The property of high dimensionality impacts the process of multivariate time series feature extraction. Traditional methods such as principal component analysis (PCA) have some limitations in representing multivariate time series [15]. A representative set is defined as a special subset of an original dataset which satisfies three main characteristics [16]. The model trained by the representative dataset can perform better than models trained by the original dataset. It shows that the feature representative matrix can help select the training data and improve the model performance.

In this study, we propose a training data selection method for a deep learning model to predict anomalies. The proposed method can be used to improve the performance of the deep learning model for the real production case dataset [3,17].

The rest of this article is organized as follows. Section 2 explains the overall procedure of our proposed method which is about how to select the training dataset based on different categorical variables in multivariate time series. Section 3 gives a real production case study and evaluation of the performance of the proposed method. Finally, we conclude and discuss our research work and give the future direction in Section 4.

2. The Proposed Method

In this section, the proposed method is presented. In Section 2.1, the basic notations and definitions used in this paper is set up. In Section 2.2, we show the flowchart of overall procedures of the proposed algorithm in this study. In Section 2.3, we give details of the procedures of the method. Further information is provided in the following section.

2.1. Basic Notations

The basic notations and definitions are described in Table 1.

Table 1. The Description of the Notations.

Notation	Description
X	A multivariate time series
D	Multivariate time series dataset
C_{ov}	The covariance matrix
λ	The eigenvalue of covariance matrix
U	The eigenvector matrix of covariance matrix
k	The number of selected features
F	The feature matrix
R	Integrated feature matrix
M	Distance matrix

(1) A multivariate time series can be denoted as:

$$\mathrm{X} = (\mathrm{x}_1(\mathrm{t}),\ \mathrm{x}_2(\mathrm{t}), \cdots, \mathrm{x}_\mathrm{m}(\mathrm{t})) \tag{1}$$

where $t = 1, 2, \cdots, n$. The size of matrix X is $n \times m$ and the matrix can be represented by:

$$X = \begin{pmatrix} x_1(1) & x_2(1) & \cdots & x_m(1) \\ x_1(2) & x_2(2) & \cdots & x_m(2) \\ \vdots & \vdots & \cdots & \vdots \\ x_1(n) & x_2(n) & \cdots & x_m(n) \end{pmatrix} \tag{2}$$

n is the time dimension which is considered the length of the multivariate time series. m is the variable dimension which stands for the number of sensors. Each column of matrix X represents a univariate time series, each row represents a group of observed values for a special time.

(2) A multivariate time series dataset can be denoted as:

$$D = \{ D_1, D_2, \ldots, D_L \} \tag{3}$$

$D_i = X^i = \left(x_1^i(t), x_2^i(t), \ldots, x_m^i(t)\right)$, where $i = 1,\ 2,\ \ldots,\ L$; X^i is a set of m univariate time series with time length t. For any i and j, D_i and D_j represent for m-dimension time series with time length $t = n_i$ and $t = n_j$, D_i and D_j can be any length, $n_i \neq n_j$ or $n_i = n_j$.

(3) The covariance matrix of X can be computed by

$$C_{ov}(\mathrm{X}) = E[(X - \overline{X})(X - \overline{X})^T] \tag{4}$$

That is

$$C_{ov}(X) = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_{m1} & \sigma_{m2} & \cdots & \sigma_{mm} \end{pmatrix} \tag{5}$$

σ_{ij} is the covariance between the ith variable and the jth variable in X, $C_{ov}(X)$ is a symmetric matrix, that is $\sigma_{ij} = \sigma_{ji}$.

(4) Through executing PCA on each covariance matrix to get the eigenvector matrix and eigenvalues, it can obtain the eigenvalues $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_m]$ and eigenvectors $U = \{U_1, U_2, \ldots, U_m\}$, where $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$. According to the value of λ_i, the information contribution of the ith principal component and the order of the information contribution can be known.

Thus, the multivariate time series D_i can be represented by the feature matrix as

$$F_i = F((i-1)m + 1 : im, :) \tag{6}$$

The first k feature often selected to represent the original matrix is $i = k$. The feature matrix is $F_{s \times s}$, $s = mL$. $F = (F_1, F_2, \ldots, F_{mL})$.

(5) After choosing the first k features, D_i can be represented by the integrated feature matrix

$$R_i = F((i-1)m + 1 : im, 1 : k) \tag{7}$$

The multivariate time series dataset can be represented by the integrated feature matrix $R = (R_1, R_2, \ldots, R_L)$, where $i = 1, 2, \ldots, L$.

(6) The Euclidean distance matrix of the integrated feature matrix R^T can be represented as:

$$M = (d_{ij}) \tag{8}$$

$d_{ij} = ||x_i - x_j||^2$; where $||\cdot||$ denotes the Euclidean norm on R^T

$$M = \begin{bmatrix} 0 & d_{12} & d_{13} & \ldots & d_{1n} \\ d_{21} & 0 & d_{23} & \ldots & d_{2n} \\ d_{31} & d_{32} & 0 & \ldots & d_{3n} \\ & & \cdots & & \\ d_{n1} & d_{n2} & d_{n3} & \ldots & 0 \end{bmatrix} \tag{9}$$

2.2. The Flowchart of Overall Procedures of the Proposed Method

Overall procedures of the proposed algorithm in this study are presented in Figure 1 and the details of the flowchart are explained.

First, the original dataset is spilt into different lengths of time series according to different categorical variables. The categorical variables in practice often represents different products in a multiple products production line. The number of the small datasets is equal to the number of categorical variables. Then, the data are spilt into positive and negative data, "positive" stands for normal behavior and "negative" stands for abnormal behavior. All the positive (y = 1) data are removed and the negative (y = 0) data are left. After the standardization, the covariance matrix of each dataset is calculated. The mean of the integrated covariance matrix is removed and PCA is applied to get the first k feature representation. The feature matrix is composed together to get integrated feature matrix. After obtaining the integrated feature matrix, the Euclidean distance matrix can be calculated. The different time length data samples can be represented by this distance matrix. The (dis)similarity can be inferred from the distance directly. Hierarchical clustering can show the clusters intuitively.

Second, the original dataset is spilt again and the training data are selected to train autoencoders. They are separated into training data and test data. The semi-supervised

training skills are used in the training process. At the end, the autoencoders have similar structures but different reconstruction errors. That means the prediction models have similar structures with different thresholds.

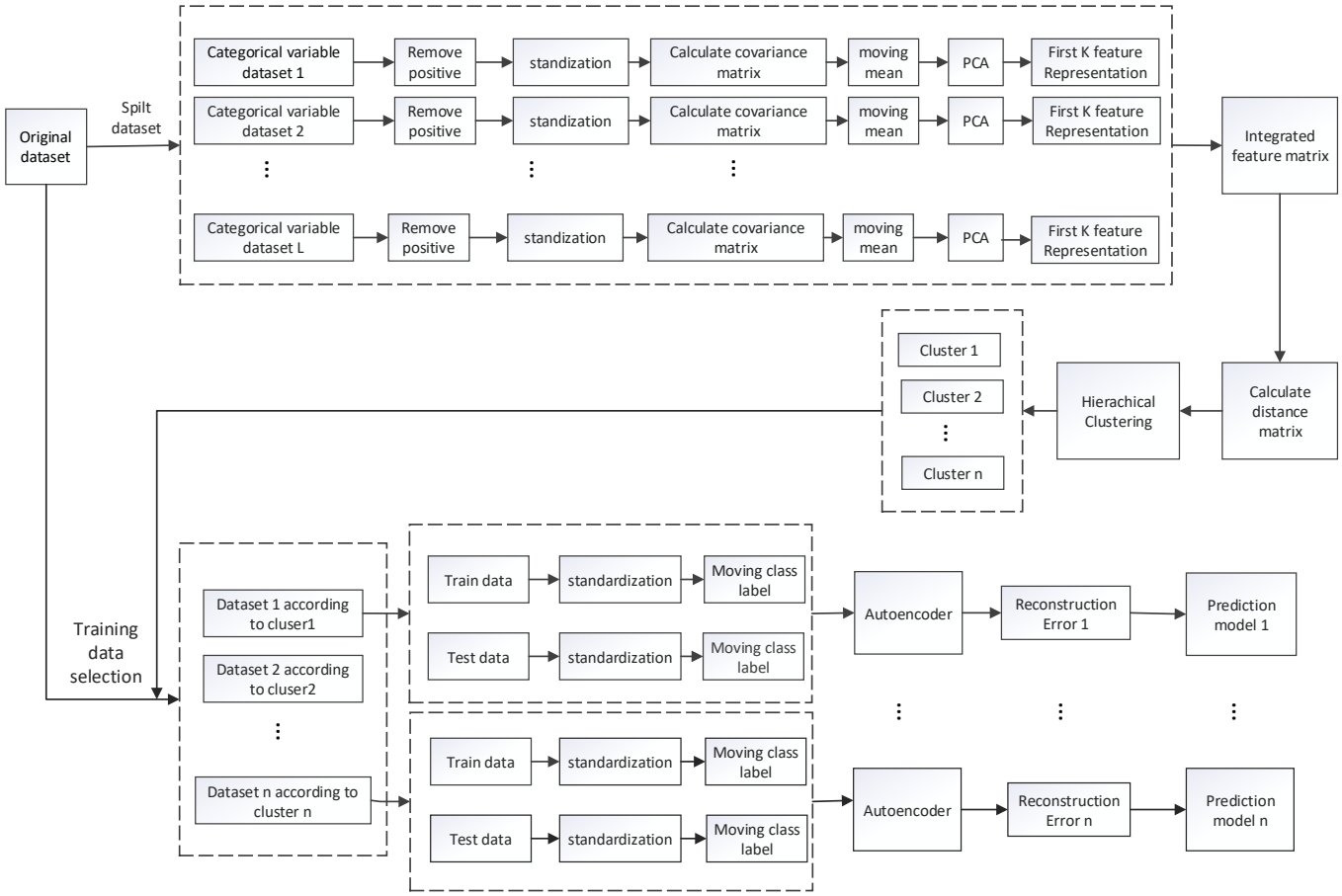

Figure 1. The flowchart of overall procedures.

2.3. The Description of the Procedures in the Proposed Method

Algorithm: Selecting the training datasets to learn the prediction models

Input: multivariate time series dataset D

Output: prediction models

Step 1. Spilt the original multivariate time series dataset according to the categorical variables. Then, we can get different time length multivariate time series samples D_i.

Step 2. Remove all the positive data which are labeled y = 1 in every data sample. All the remaining data are negative data which are labeled y = 0. They are considered as normal behavior data.

Step 3. Standardized multivariate time series D_i, eliminate the scale influence between the components. $D_i = zscore(D_i)$ for any variable x_i, the standardization is $x_i = (x_i - u_i)/\sigma_i$, where u_i and σ_i represent the mean and standard deviation of univariate time series x_i.

Step 4. Calculate the covariance matrix C_i of every data sample. Any two multivariate time series with different lengths have the equal-length covariance matrix.

Step 5. Delete the mean value for all the columns of C_i, $C_i = C_i - \overline{C_i}$, $\overline{C_i}$ stands for the mean of C_i.

Step 6. Apply PCA on the covariance matrix to get the eigenvector matrix and the eigenvalues. Choose the first k eigenvectors as the coordinate axes of the new system according to Equation (5).

Step 7. Let the k eigenvectors be the features matrix F of the multivariate time series dataset according to Equation (6). In our case, we choose k = 2. It can retain 99.92% of the information of each covariance matrix.

Step 8. Integrate the feature matrix F into a representation matrix R according to Equation (7). That is a representative set. In this way, the dataset D is represented by an integrated feature matrix.

Step 9. Calculate the distance matrix between every two multivariate time series according to Equation (9), that is the Euclidean distance matrix of M^T.

Step 10. Do hierarchical clustering to cluster them into groups [18]. The hierarchy represents an ordered sequence of groupings. The measure of dissimilarity between sets of observations is achieved by the pairwise distance matrices The dissimilarity between clusters increases with the level of the merger. Select the similar mode normal data which have similar distance.

Step 11. Select the similar normal data as input training data for reconstruction anomaly detection model. An autoencoder can be used as a reconstruction model which attempts to reconstruct its inputs from itself [18,19]. In our case, autoencoder is used for semi-supervised learning as anomaly detection model.

Step 12. Get all the parameters of the model, especially set the construction error as a threshold. A well-trained reconstruction model will be able to accurately reconstruct a new sample. The inferences for rare event prediction are made by classifying the reconstruction errors as high or low. Multiple models with different thresholds are trained.

Steps 1 to 10 are the proposed training data selection method based on feature representation matrix calculated by the categorical variables in the multivariate time series data. Steps 11 to 12 are the regular training deep learning model processes.

3. A Real Case Study and Results

In this section, we apply the proposed method to a real case study. We first give a description of the challenges and difficulties of a critical problem in pulp and paper production. Then, we describe a real dataset from a paper manufacturing process which is a multiple products production line. Finally, we apply our method to analyze the dataset and explain the results.

3.1. The Description of the Critical Problem in Pulp and Paper Production

Pulp and paper production is one of the largest manufacturing sectors in the world which has increased globally and will continue to increase in the near future. Pulp and paper production requires highly complex and integrated processes by chemical or mechanical means, which include wood preparation, pulping, chemical recovery, bleaching, and paper making. There are many sensors located along the production line and the data are collected from the start to the end products [12,17].

Paper manufacturing is a continuous process. However, on average, a mill witnesses more than one sheet break every day. Each sheet break can cause downtime of an hour or longer which is a critical problem in the production. These sheet breaks are unwanted and the downtime causes the loss of millions of dollars at a plant and billions across the industry. Even a small reduction in these breaks would lead to significant savings. More importantly, fixing a sheet break often requires an operator to enter the paper machine. These are large machines with some hazardous sections that pose a danger to operators' health. Preventing sheet break via predictive systems will make operators' work conditions better and the paper production more sustainable.

There are many challenges and difficulties of this problem. The cause of paper sheet break is the problem which is usually instant. The sheet breaks occur infrequently, so they are rare events. Predicting such events before they occur is thus extremely challenging. The objective of the problem is early detection of an anomaly or failure.

3.2. The Description of the Original Dataset

A real dataset from a pulp-and-paper mill with a break failure was provided [3,17]. The original data can be seen as multivariate time series. In Figure 2, 6 variables are selected from 61 variables to show original data collected from the production. These data were collected by the sensors placed in different parts of the paper machine along its length and breadth. They can be also seen as multi-sensor stream data. The paper machines are typically several meters long and ingest raw materials at one end and produce reels of paper at the other end. These sensors measure both raw materials and process variables.

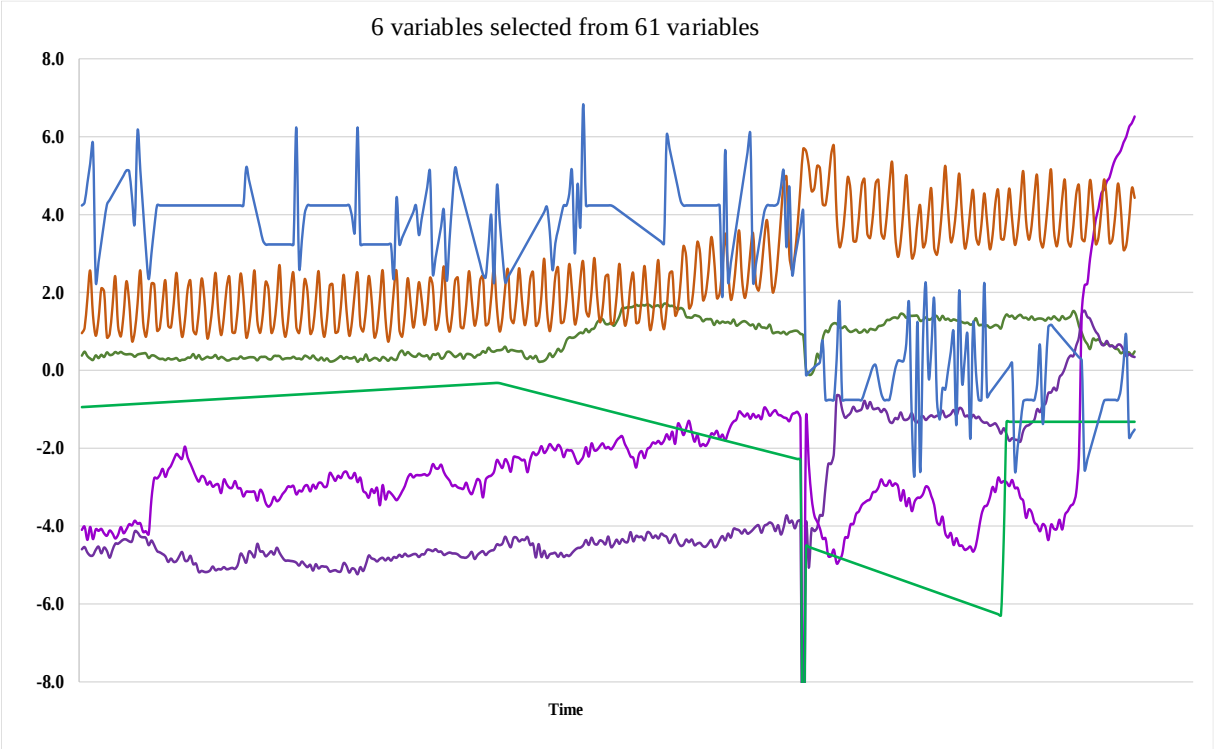

Figure 2. Original multivariate time series (6 variables selected) from manufacturing process.

The data collected by the sensors are multivariate time series. There are 18,398 rows and 62 columns. There are 61 variables and two of them are categorical variables. The categorical variable ×28 stands for the paper grade. The dataset is extremely imbalanced data since the ratio of positive data is 0.67%. Statistically, if an event constitutes less than 5% of the dataset, it is categorized as a rare event. In this case study, the breaks which occur at less than 1% are discussed and modeled. The basic description of the spilt data samples is in Table 2. The variable ×28 has eight categorical numbers which stand for different paper grades. The total failures amount to 124.

Table 2. The statistics of original dataset.

Rank	×28	Rows	Anomaly	Ratio
1	96	6574	72	0.0110
2	82	4378	18	0.0041
3	118	2646	15	0.0057
4	139	1807	10	0.0055
5	112	1235	5	0.0040
6	84	1313	2	0.0015
7	93	419	1	0.0024
8	51	26	1	0.0385
		18,398	124	0.0067

3.3. The Result of the Method

The proposed method was applied to the multivariate time series data to split the original data and get groups of the paper grades according to the results of hierarchical clustering. According to the clustering results, different training datasets were selected to train different prediction models.

3.3.1. The Result of the Distance Matrix and Hierarchical Clustering

Table 3 shows the distance matrix of each normal data sample. The distance of all the paper grades is around 0.2 except paper grade 51. Paper grades 1–7 can be clustered together. The distance shows that the normal production behavior of paper grade 51 is different from the others. This means that paper grade 51 has a big difference compared with other paper grades. It would be better to not train the data together because paper grade 51 may make the autoencoder struggle to learn the parameters and cause a big reconstruction error. This can be known from the clustering results shown in Figure 3 as well.

Table 3. The distance matrix of each normal dataset (k = 2).

Rank	1	2	3	4	5	6	7	8
1	0.000	0.202	0.195	0.172	0.213	0.197	0.221	**0.827**
2	0.202	0.000	0.185	0.180	0.275	0.258	0.248	**0.847**
3	0.195	0.185	0.000	0.209	0.244	0.241	0.265	**0.805**
4	0.172	0.180	0.209	0.000	0.184	0.177	0.212	**0.827**
5	0.213	0.275	0.244	0.184	0.000	0.114	0.174	**0.810**
6	0.197	0.258	0.241	0.114	0.114	0.000	0.191	**0.789**
7	0.221	0.248	0.265	0.212	0.174	0.191	0.000	**0.834**
8	**0.827**	**0.847**	**0.805**	**0.827**	**0.810**	**0.789**	**0.834**	**0.000**

Note: 1–8 stand for 96, 82, 118, 139, 112, 84, 93, 51.

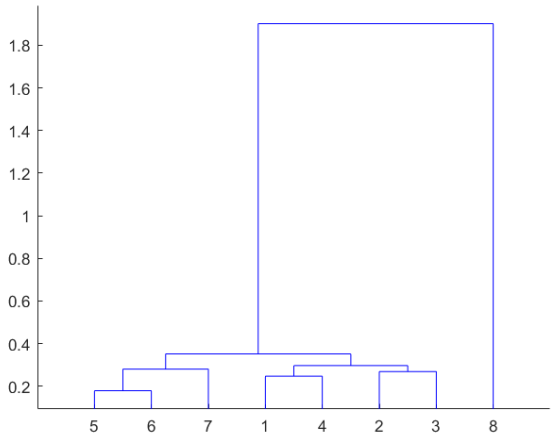

Figure 3. The result of hierarchical clustering.

Figure 3 shows the results of hierarchical clustering of the distance matrix. The results show that the total data can be clustered into two clusters. The paper grade 51 data are in one group and the other paper grades are in the other group.

3.3.2. The Performance of Autoencoders after Selecting the Training Data

A training dataset is a special set of labeled data providing known information that is used in the supervised learning or semi-supervised learning to build a classification or regression model. In our case, the training dataset was used to learn an anomaly detection or prediction model. The subset of the data contains all the information necessary to build an autoencoder model. The goal of the training phase was to estimate parameters of a model to predict output values with a good predictive performance in real use of the model.

(1) Training procedure

After hierarchical clustering, the categorical variables could be clustered into two groups. We chose the similar normal state data which are the original dataset without the data of paper grade 51. First, we trained the autoencoder the same way as we trained the model with all the data. Then, we compared the prediction results and found that the model which is trained by the data without paper grade 51 performed better. In other words, training the data according to different normal state data samples works significantly better for an autoencoder.

The training procedures are presented below:

(a) Divide the process data into two parts: majority class, negatively labeled as {x, ∀t | y = 0}, and minority class, positively labeled as {x, ∀t | y = 1}.

(b) The majority class is treated as a normal state of a process. The normal state is when the process is break-less.

(c) Train a reconstruction model on the normal state samples {X, ∀t | y = 0}, i.e., ignore the positively labeled minority data.

(2) Performance analysis

It is necessary to evaluate the model we built to estimate predictive performance. Strategies and measurements for the model evaluation have been well described. Since the industrial case is an imbalanced problem, F1-score is used as the accuracy measure of evaluation. In addition to the F1_score, true positives (TP), false positives (FP), true negatives (TN), false negatives (FN), accuracy, precision, recall, and false positive rate (FPR) are reported [20].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

$$Precision: \ P = \frac{TP}{TP + FP} \tag{11}$$

$$Recall: \ R = \frac{TP}{TP + FN} \tag{12}$$

$$\text{F\_measure}: \ F = \frac{\left(a^2 + 1\right) * P * R}{a^2 * (\text{P} + \text{R})} \tag{13}$$

$$\text{F1\_score } (\text{a} = 1): \ F1 = \frac{2 * P * R}{P + R} \tag{14}$$

For a reliable future error prediction, the model needs to be evaluated on a different, independent, and identically distributed dataset which is different to the dataset used for building the model. We can split the original dataset into more subsets to simulate the effect of having more independent identically distributed datasets.

This is 4 min earlier autoencoder. The parameters are set the same as in Ranjan's work [3] to show the improvement to autoencoder by the proposed method.

Whole data are the original data of all the paper grades. Data without 51 are the original data except the data of paper grade 51. From the results of Table 4, it is clear that the false positive was reduced and true positive was improved after selecting the training dataset. The results of Table 5 show that the proposed method improved all the performance measurement. Both F1 score and precision of training after selecting the data were improved almost 20 percent.

Table 4. The performance comparison of the autoencoder results which are trained by different training data.

True\Predict	Whole Data		Data without 51			
	0	**1**	**0**	**1**		
0	2760	123	2761	117	**TN**	**FP**
1	35	6	34	8	**FN**	**TP**

Table 5. Different measurement of the performance.

	Accuracy	**Precision**	**Recall**	**F1_Score**	**Auc**	**True Positive Rate**	**False Positive Rate**
Whole data	0.946	0.046	0.146	0.07	0.707	0.146	0.043
Without 51	0.948	0.064	0.19	0.095	0.728	0.190	0.041

4. Conclusions and Discussion

In this section, we give a conclusion of the results and the advantages and disadvantages of our method.

The main contribution of this paper is designing a new way to sample the training data that can improve the deep learning model accuracy (sensitivity and recall) for multivariate time series data. The proposed method split the multivariate time series dataset into multiple small samples by categorical variables. The multiple samples stand for different products from the same production line. The dis(similarities) of these samples can be quantified by distance matrix and qualified by hierarchical clustering. The autoencoder is used as a reconstruction model and the reconstruction error is used as a threshold to detect the rare event. According to the similarities, we selected a suitable training dataset to fit the autoencoders to get a new reconstruction error as a threshold. The performance of the rare event prediction model can be improved significantly after training data selection.

It is normally very difficult to accurately determine a set of training data that is adequately representative of the target. There are few state-of-the-art approaches to focus on training data selection for us to do a comparison. The method is applied to a real complex manufacturing process. For multiple products production lines or manufacturing in practice, it is really a good way to do training data pre-selection before applying the deep learning method.

Advantages of this approach are its simplicity and very low computational complexity. The proposed method is used to sample training data to improve the autoencoder performance. Since the proposed method can obtain equal-length feature vectors, it can be used as a feature representation method or clustering method for other multivariate time series data. Multivariate time series data can be seen everywhere in engineering, which is one kind of the important data that needs to be mined for valuable information and knowledge. Reconstruction anomaly models other than autoencoders can be integrated since they share similar structures. The proposed method is easily applied in real-world applications.

The results show that too much compression will cause useful information loss. A greatly reduced dimension makes the methods have over-redundant information that causes bad results. Since the value k can be decided subjectively and it needs experience or domain knowledge to be decided carefully. A different value of k may lead to different feature representation. Different feature representation may lead to a different distance matrix and the final results of training data selection may be different. Experts are required in order to have the necessary domain expertise to obtain a good integrated feature matrix. For different data clustering results, more than one prediction models should be built.

Future work will involve doing more experiments with other deep learning models, for example, carrying out clustering experiments to compare the computation speed and the efficiency of the Euclidian distance and DTW distance.

Author Contributions: Conceptualization, methodology, validation, formal analysis, writing—original draft preparation, D.X.; writing—review and editing, Z.Z. and J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, China (Grant No. 51775108).

Data Availability Statement: The link to the dataset can be found in Ranjan's book. The dataset was retrieved from https://drive.google.com/file/d/1rmSaluLD2pAD8s183T7nKhu0O4yeYewq/view (accessed on 8 December 2020).

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

References

1. Heaven, W.D. *The Way We Train AI Is Fundamentally Flawed*; MIT Technology Review; MIT Press: Cambridge, MA, USA, 2020.
2. Redman, T.C. *If Your Data Is Bad, Your Machine Learning Tools Are Useless*; Harvard Business Review; Harvard University Press: Cambridge, MA, USA, 2018.
3. Ranjan, C. *Understanding Deep Learning and Applications on Rare Event Prediction*, 1st ed.; Connaissance Publishing: Atlanta, GA, USA, 2020.
4. Lee, W.; Seo, K. Early failure detection of paper manufacturing machinery using nearest neighbor-based feature extraction. *Eng. Rep.* **2020**, *3*, e12291. [CrossRef]
5. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, P.W. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
6. Petitjean, F.; Ketterlin, A.; Gancarski, P. A global averaging method for dynamic time warping with applications to clustering. *Pattern Recognit.* **2011**, *44*, 678–693. [CrossRef]
7. Branco, P.; Torgo, L.; Ribeiro, R.P. A survey of predictive modeling on imbalanced domains. *ACM Comput. Surv.* **2016**, *49*, 1–50. [CrossRef]
8. Nalepa, J.; Kawulok, M. Selecting training sets for support vector machines: A review. *Artif. Intell. Rev.* **2019**, *52*, 857–900. [CrossRef]
9. Dai, W.; Yoshigoe, K.; Parsley, W. Improving data quality through deep learning and statistical models. In *Information Technology-New Generations*; Springer: Cham, Switzerland, 2018; pp. 515–522.
10. Borghesi, A.; Bartolini, A.; Lombardi, M.; Milano, M.; Benini, L. Anomaly Detection Using Autoencoders in High Performance Systems. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; AAAI Press: Palo Alto, CA, USA, 2019; Volume 33, pp. 9428–9433.
11. Bonissone, P.P.; Goebel, K. When will it break? A hubrid soft computing model to predict time-to-break margines in paper machines. In Proceedings of the SPIE 47th Annual Meeting, International Symposium on Optical Science and Technology, Seattle, WA, USA, 7–11 July 2002; pp. 53–64.
12. Bissessur, Y.; Martin, E.B.; Morris, A.J. Monitoring the performance of the paper making process. *Control. Eng. Pract.* **1999**, *7*, 1357–1368. [CrossRef]
13. Singhal, A.; Seborg, D. Clustering multivariate time-series data. *J. Chemom.* **2005**, *19*, 427–438. [CrossRef]
14. Keogh, E.; Ratanamahatana, C. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.* **2004**, *7*, 358–386. [CrossRef]
15. Pearson, K. On lines and planes of closest fit to systems of points in space. *Philos. Mag.* **1901**, *2*, 559–572. [CrossRef]
16. Borovicka, T.; Jirina, M., Jr.; Kordik, P.; Jirina, M. Chapter 2: Selecting Representative Data Sets. In *Advances in Data Mining Knowledge Discovery and Applications*; InTechOpen: London, UK, 2012. [CrossRef]
17. Ranjan, C.; Mustonen, M.; Paynabar, K.; Pourak, K. Dataset: Rare event classification in multivariate time series. *arXiv* **2018**, arXiv:1809.10717v2.
18. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer: Cham, Switzerland, 2009. [CrossRef]
19. Theodoridis, S. *Machine Learning: A Bayesian and Optimization Perspective*; China Machine Press: Beijing, China, 2017.
20. Sokolova, M.; Japkowicz, N.; Szpakowicz, S. Beyond Accuracy, F-score and ROC: A family of Discriminant Measures for performance Evaluation. In *AI 2006: Advances in Artificial Intelligence, Proceedings of the Australasian Joint Conference on Artificial Intelligence, Hobart, Australia, 4–8 December 2006*; Springer: Berlin/Heidelberg, Germany, 2006. [CrossRef]