Hindawi
Advances in Multimedia
Volume 2023, Article ID 9802464, 1 page
https://doi.org/10.1155/2023/9802464

Retraction

Retracted: Exploring Artificial Intelligence Architecture in Data Cleaning Based on Bayesian Networks

Advances in Multimedia

Received 12 December 2023; Accepted 12 December 2023; Published 13 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] S. Zhang, Y. Wang, and Q. Lv, "Exploring Artificial Intelligence Architecture in Data Cleaning Based on Bayesian Networks," *Advances in Multimedia*, vol. 2022, Article ID 6731781, 11 pages, 2022.

Hindawi
Advances in Multimedia
Volume 2022, Article ID 6731781, 11 pages
https://doi.org/10.1155/2022/6731781

Research Article

Exploring Artificial Intelligence Architecture in Data Cleaning Based on Bayesian Networks

Suzhen Zhang, Yuechun Wang, and Qing Lv

Shijiazhuang Posts and Telecommunications Technical College, Shijiazhuang 050021, China

Correspondence should be addressed to Yuechun Wang; wangyc_202205@163.com

Received 15 July 2022; Revised 5 August 2022; Accepted 20 August 2022; Published 13 September 2022

Academic Editor: Tao Zhou

In order to further improve the technical level of data cleaning and data mining and better avoid the defects of uncertain knowledge expression in traditional Bayesian networks, a Bayesian network algorithm based on combined data cleaning and mining technology is proposed, and a manual functional data cleaning architecture based on Hadoop is constructed. The results show that the traditional neighbor sorting algorithm with window size of 5 takes the least time to process the same amount of data. The nearest neighbor sorting algorithm with window size 7 is always the longest. The time consumption of the nonfixed window nearest neighbor sorting algorithm is similar to that of the traditional nearest neighbor sorting algorithm with a window size of 5. However, with the increase of data volume, the consumption time increases rapidly until it approaches the consumption time of the traditional sorting nearest neighbor algorithm with window size of 7. Therefore, the algorithm can improve the precision of data cleaning at the expense of cleaning speed, which proves that the artificial intelligence architecture based on combined data significantly improves the efficiency of the algorithm and can effectively analyze and process large data sets.

1. Introduction

The emergence of the Big Age has almost completely changed the outlook for many industries. In addition, the emergence and popularity of intelligent technology have made data storage and data mining even more important; in such cases, data cleaning improves the good information data in big data and helps businesses and users to get more important information. [1]. Therefore, in order to improve the efficiency and quality of data maintenance based on Bayesian networks, this article has established a Bayesian networking organization algorithm based on a combination of data cleaning and deleting technology and manual data cleaning. Hadoop's to improve the performance of big data.

2. Literature Review

Feng, L. believed that foreign research on data laundering, such as by adjusting the security number in the USA, first appeared in the USA. In the USA, data and business development has boosted research in this area of technology. Research is focused on the detection and removal of abnormal data [2]. Yang, Z. believes that for numerical attributes, abnormal data can be detected and eliminated by comparing the set confidence interval with statistical methods of calculating field mean and standard deviation. For character attributes, common methods include data mining for anomaly detection, clustering based editing distance method, pattern learning based method, and association rule method [3]. Wu, H. D. considers identifying and removing similar information. This procedure is to remove similar equations. The search and removal of similar data is a research focus on data cleaning [4]. Zhou, Y. believes that in the process of integrating different systems, there will be a large number of similar and repeated records, and these information may appear redundant or even contradictory. Field matching and record matching are commonly used to determine whether two records are similar and duplicate [5]. Wu, L. believes that currently, the more commonly used field matching algorithms include basic field matching algorithm, recursive field matching algorithm, and algorithm. The commonly used record matching algorithms are basic neighbor

sorting algorithm its several improved algorithms, such as multineighbor sorting method and priority queue algorithm. Data integration is the primary consideration of data cleaning in data warehouse. It is mainly to map the structure and data in data source to the target structure and domain [6].

Scanagatta, M. believes that the research on data cleaning started late in China, especially the linguistic differences between English and Chinese, resulting in the inability of English-based data cleaning technology to be fully applicable to the situation in China [7]. Costa, G. believes that in recent years, more and more attention has been paid to the research of data cleaning technology in China, and some achievements have been made. For example, the similar duplicate record detection method and the comprehensive multilanguage data duplicate record detection method proposed by the professor team of Fudan University are relatively efficient methods to detect and eliminate similar duplicate records. In terms of data cleaning in the process of data integration, Professor Qing's team from Peking University solved the problem of data cleaning in the process of data conversion [8]. Iwakami, Y. believes that due to the lack of theoretical research on Chinese data cleaning, Chinese data cleaning tools are rarely seen in the market and seldom applied to engineering projects, resulting in the current situation of single data cleaning function, poor scalability, and universality in engineering. In general, the research on Chinese data cleaning in China is still in the initial stage [9].

3. Related Theories and Techniques

3.1. Bayes' Theorem. In terms of the result of something happening, the base θ parameter is considered a random variable, not a fixed number. In the absence of events, θ parameter can be obtained from the previous distribution, and if there is a new condition, θ can be further modified as received. Thus, the occurrence of random events A and B can be calculated. The causal event is for event A and event B, and the event's $P(A|B)$ indicates the probability that incident A will occur in event B. Co-probability: The combination can be represented by $P(A, B)$ or $P(\mathrm{AB})$ [10], the probability that event A will occur, such as event B. Edge event: The edge result is the result of a single event A or only B, which can be expressed as $P(A)$ and $P(B)$.

Bayes' theorem can be derived from the above relevant knowledge, and the joint probability can be expressed as

$$P(A, B) = P(A|B)P(B) = P(B|A)P(A). \quad (1)$$

When on the right side of the equation is divided by $P(B)$, Bayes' theorem can be obtained, that is, the expression of the conditional probability $P(A|B)$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}. \quad (2)$$

If it is difficult to directly calculate the probability of $P(B)$ for event B, the method of total probability can be used to calculate it. Event B is decomposed into several small events, and the probability of event B is obtained by summing the probabilities of the small events. When event B is decomposed, event B is not decomposed directly, but the sample space is divided into $A1, A2, A3, \cdots,$ An event, and $A1, A2, A3, \cdots,$ An constitutes a complete event; that is, two of them do not intersect each other, and their sum is universal [11]. Thus, event B can be defined by events BA1 , BA2, $\cdots,$ BAn. When each event A_i occurs leading to the probability of event B occurs is $P(B|A_i)$, event B can be expressed as

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i). \quad (3)$$

Then, Bayes' theorem can be expressed as

$$P(A_i|B) = \frac{P(A_i)P(B|A_i)}{\sum_j^n P(A_j)P(B|A_j)}, \quad (4)$$

In the above formula, A_i is the cause of event B, and $P(A_i)$ represents the probability of occurrence of various causes, namely, prior probability or edge probability. $P|B(A_i)$ is that after event A occurs, the prior probability is modified, which is the posterior probability, or the conditional probability.

3.2. Bayesian Network. Bayesian network consists of directed acyclic graph and conditional probability table, as shown in Figure 1. Where directed acyclic graph is denoised as $G(V, E)$, node V in the graph represents random variable, and edge E represents the dependence between nodes. For example, $V1 \longrightarrow V2$ indicates that the occurrence of node $V1$ will lead to the occurrence of node $V2$. $V1, V2 \longrightarrow V3$, indicating that the simultaneous occurrence of node $V1$, and $V2$ leads to the occurrence of node 3. Bayesian network considers that the current node is only dependent on its parent node and is independent of other node conditions. The conditional probability table stores the probability of occurrence of the current node under different parent nodes, as shown in Tables 1, 2, and 3. The strength of dependence between nodes can be expressed through conditional probability [12]. Variables in Bayesian networks can be either continuous variables or discrete variables. This paper mainly discusses Bayesian network learning about discrete variables.

The probability of Bayesian networks can calculate the probability of any sample point in a discrete space. The characteristic of Bayesian networks is that they explicitly express conditional independence and dependence between events. The joint probabilities of all N variables in the network can be obtained by multiplying their local conditional probabilities, as shown in the following formula:

$$P(X_1, \cdots, X_n) = \prod_{i=1}^{n} P(X_i|\pi_i). \quad (5)$$

Therefore, any form of $P(A|B)$ probability can be determined, where A and B are sets of variables with known values.

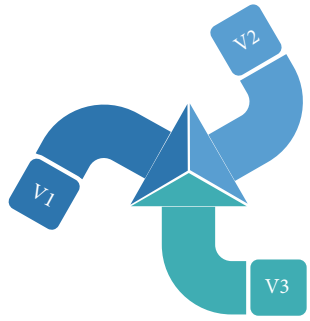

Figure 1: Bayesian network G.

Table 1: Conditional probability table of node $V1$.

True	False
0.4	0.6

Table 2: Conditional probability table of node $V2$.

$V1$	$V2$ = True	$V2$ = False
True	0.6	0.4
False	0.5	0.5

Table 3: Node 3 conditional probability table.

$V1$	$V2$	$V3$ = True	$V3$ = False
True	True	0.2	0.8
True	False	0.3	0.7
False	True	0.9	0.1
False	False	0.4	0.6

3.3. Bayesian Network Learning

3.3.1. Bayesian Network Structure Learning. At present, Bayesian structure learning is divided into dependency based Bayesian network structure learning and scoring based Bayesian network structure learning [13]. (1) The method based on the dependence relationship is mainly to judge the dependence relationship between variables through the conditional independence. If two variables are independent of each other, there will be no dependence relationship; otherwise, there will be, and the method to judge the dependence relationship between variables is mainly mutual information or conditional mutual information. (2) Based on the scoring search method, this method mainly scores the network structure searched to learn the Bayesian network.

This paper mainly uses the second method to construct Bayesian network, and $K2$ is a classical scoring search algorithm. After a given database D, the algorithm is used to search with maximum probability $P(G)|DG$ of Bayesian network structure. Assuming two network structures, $G1$ and $G2$, the scoring function is used to compare the probability of learning $G1$ and $G2$ structure for database D. As can be seen from Formula (6), it is to find a method to calculate $P(G, D)$ [14]. In a given G Bayesian network structure, and assuming that the independent occurrences and conditional probability density function $f(\mathrm{GP}|G)$ are uniform, Formula (7) can be obtained by derivation.

$$\frac{P(G1)}{P(G2)} = \frac{P(G1, D)}{P(G2, D)}, \tag{6}$$

$$P(G, D) = P(G)\prod_{i=1}^{n}\prod_{j=1}^{q_i}\frac{(r_i - 1)!}{(N_{\mathrm{ij}} + r_i - 1)!}\prod_{k=1}^{r_i}N_{\mathrm{ijk}}!. \tag{7}$$

where N is the number of nodes; r_i is the number of random variable X_i; Q_i is the number of values of the parent node set $\pi(I)$ of the random variable X_i; N_{ijk} is the number of samples in which the random variable X_i takes the KTH value and the parent node takes the JTH value. By $K2$ algorithm, assuming that each network structure G is equally possible, then $P(G)$ can be regarded as a constant and adopts greedy method to maximize $P(G, D)$. Each node and its parent node set are regarded as a local structure, and Equation (8) can be regarded as the product of scores of n local structures of equation; that is, each local structure can be calculated by the following equation:

$$g(X_i, \pi(i)) = \prod_{j=1}^{q_i}\frac{(r_i - 1)!}{(N_{\mathrm{ij}} + r_i - 1)!}\prod_{k=1}^{r_i}N_{\mathrm{ijk}}!. \tag{8}$$

The $K2$ algorithm starts by assuming that the node lacks a parent, and then, at each step it adds a parent that increases the value of $g(X_i, \pi(I))$.$K2$ stops adding the parent set to the node when the added parent can no longer increase $g(X_i, \pi(I))$. $K2$ algorithm can obtain relatively accurate Bayesian network structure by learning in big data sets. However, it can be seen from the above description that $K2$ algorithm uses scoring function to calculate each searched structure, with high calculation intensity and low learning efficiency in big data sets [15]. Aiming at this problem, this paper improves the $K2$ algorithm by reducing the $K2$ scoring function, reducing the calculation intensity and improving its learning efficiency.

3.4. Bayesian Network Parameter Learning. Bayesian network parameter learning is to learn the conditional probability table of each node when the structure of Bayesian network is known. The classic method of parameter learning is maximum likelihood estimation algorithm, which is also used in this paper for parameter learning. The following focuses on the method of maximum likelihood estimation. Maximum likelihood estimation takes likelihood function as optimization objective, and the process of parameter estimation can be regarded as optimization process. This parameter is the most likely set of parameters if it makes the data most likely to occur. The likelihood function obtained by discrete random function is the probability of getting this sample. Since each sample is independent, the probability of the sample can be obtained by multiplying

the probability of each sample, which is the likelihood function [16]. When a given output x, for likelihood function of parameter theta $L(\text{theta}\,|x)$, theta is equal to the given parameters in numerical variable x probability:

$$L(\theta|x) = P(X = x|\theta). \tag{9}$$

Let us take a simple example. First, assume that the random variable X is a trivariate distribution, and the variables are randomly 0, 1, 2. The probability distribution of X is shown in Table 4:

The unknown quantity is in it. The sample value obtained by observing X is $\{0, 1, 2, 0, 2, 1\}$.

$$L(\theta) = \theta(1-2\theta)\theta\theta(1-2\theta) = \theta^4(1-2\theta)^2. \tag{10}$$

So the logarithm function is zero.

$$L(\theta) = 4\ln\theta + 2\ln(1-2\theta). \tag{11}$$

Finally, the maximum likelihood estimate is 1/3. The above method is extended to Bayesian networks, one consisting of N variables:

$$X = \{X_1, X_2, \cdots, X_n\}. \tag{12}$$

A Bayesian network B:

$$\theta = \{\theta_1, \theta_2, \cdots, \theta_n\}. \tag{13}$$

It is a set of conditional probability distribution tables, where

$$\theta_i, i = 1, 2, \cdots, n. \tag{14}$$

The above equation is the conditional probability distribution table of random variable X_i, and then, each term in θ is

$$\theta_{\text{ijk}} = P(X_i = k|\pi(X_i) = j). \tag{15}$$

When Formula (16) is satisfied, the likelihood function is maximum.

$$\theta_{\text{ijk}} = \frac{m_{\text{ijk}}}{\sum_{k=1}^{r_i} m_{\text{ijk}}}, \tag{16}$$

where m_{ijk} represents the number of samples when the random variable takes the first value k in the data and its parent node takes a value j. The conditional probability parameters of nodes can be obtained by calculating the frequency of different values of nodes when the values of the given parent node set are calculated [17].

4. Hadoop-Based Manual Functional Data Cleaning Architecture

4.1. Introduction to Experimental Data. The data set of this paper is the user click log data of cloud themed products of a website. The data set contains more than 1.4 million click logs during the campaign. At the same time, it provides the corresponding relationship between commodities and themes and the purchase logs of users in the month before the activity. Based on the above situation, this paper uses relevant technologies of big data Hadoop platform and data cleaning algorithm to clean log data.

Table 4: Probability distribution.

X	O	1	2
P	θ	1-2θ	θ

4.1.1. Data Set Information. The data set contains fields and definitions as shown in Table 5.

4.1.2. Data Set Characteristics. Log data is included in the category of big data, and the 5V features of big data are also the general feature of log data. The five V characteristics of big data, respectively, are as follows:

(1) Volume: According to IDC, up to 8ZB of data are generated globally every year

(2) Varity: Common texts, images, audio and logs, videos, and blogs

(3) Velocity: The speed of massive data leads to the rapid growth of data

(4) Value: Low value density. For example, only one or two frames of a video may be useful

(5) Veracity: Four other features of data determine the uncertainty of data

Log big data not only meets the 5V features of big data, but also because of the uncertainty and periodicity of the network itself, as well as the multiple sources, wide range, and contents of log data, no special cleaning protocol can be used. Log data has its own unique characteristics:

(1) Large amount of data. Log data is generated whenever a user accesses it 24 hours a day. And there is the backup data of the server itself. Moreover, the amount of data is huge, and the growth rate of data is extremely fast

(2) Universality. Log is to record the website, software, and other basic information carrier

(3) Variability. The purchasing power of users varies with seasonal and climatic changes

(4) Diversity. Different companies, different products, different formats stored in servers, different attributes, and different record names, result in data diversity because this information can be set artificially [18]

Based on the above analysis, it can be seen that the main causes of dirty data in log data are as follows: hardware and

Table 5: Fields contained in data set.

The field name	The field type	Field meaning
user_ id	String	User ID after MD5
theme_ id	String	ThemeID
item_ id	String	Item ID after MD5
clk_ cnt	int	The number of times a user clicks on a product on the current page
cate_ levell_ id	String	ID of the highest-level category corresponding to an ITEM after MD5
leaf_ cate_ id	String	Lowest level category ID of an ITEM after MD5
reach_ time	String	User arrival time on the page

Table 6: Host name settings.

The name of the machine	The IP address
centos 1	192.168. 175.129
centos 2	192.168. 175.130
centos 3	192.168.175.131

Table 7: Configuration of Zookeeper storage parameters.

The device name	Storage parameter configuration
server.1	192.1 68.175. 129:2888:3888
server.2	192.1 68. 175.130:2888:3888
server.3	192.1 68.175. 131:2888:3888

network transmission. If the network signal is poor, some records may be lost; data comes from multiple data sources; and the same data may exist in multiple systems. Inconsistent storage standards in different systems will lead to repeated data and inconsistent data in the process of system data integration. Duplicate records cause data redundancy, waste of storage space, and bandwidth consumption, and affect the effect of subsequent data processing. The log generation process determines less inconsistent data than other types of data. Therefore, the main problem existing in the subsequent use of log data is data duplication [19].

4.2. Experimental Environment Construction. Hadoop cluster is running on Linux system, using CentOS7 version. Hadoop version is Hadoop2.7. Zookeeper is installed first. Due to the half-alive, half-agreed, and half-elected mechanisms of Zookeeper, cluster deployment uses three VMS to build a fully distributed Hadoop cluster. A Hadoop cluster consists of Namenode nodes and Datanodes. Namenode stores metadata and manages the running of Datanodes. The other two VMS are Datanodes, which store real data in blocks. The host name settings are shown in Table 6.

A Hadoop cluster needs to be managed by a Zookeeper cluster. Therefore, a Zookeeper cluster needs to be set up first, and then, a Hadoop cluster needs to be set up after the Zookeeper cluster is successfully set up.

(1) The process for setting up a Zookeeper cluster environment is as follows:

Upload the Zookeeper installation package to the Linux operating system and decompress it. CFG file in the conf directory of the Zookeeper installation directory, copy the parameter configuration template from zoo_sample. This directory is most likely to be empty, so change it. At the same time, add the following configuration to the configuration file: Set 2888 as the atomic broadcast port and 3888 as the election port [20] (see Table 7).

In the directory/home/work/zkdata, create the myID file. The myids of centos1, Centos2, and CentoS3 are 1, 2, and 3, respectively. Assign serial numbers to each node in the cluster for ease of management. Configure other nodes in the cluster: Run the scp-r command to remotely copy the Zookeeper installation package of CENtos1 and all configuration files of the current VM to the corresponding directories of Centos2 and CentoS3 VMS. Change the node IP and myID in each virtual machine. Start Zookeeper: Go to the bin directory and run the start command [21].

(2) The process for setting up a Hadoop cluster environment is as follows:

(i) Configure the host file: corresponding to the above host name

(ii) Install JDK: JDK1.8. Edit the/etc/profile file: mainly set the path of the JDK. After editing, use the source/etc/profile command to make the configuration changes take effect immediately. Use the java-version command to view the JDK version information. If 1.8.0 is displayed, the configuration is successful

(iii) Configure password-free SSH login

To facilitate the primary node to control other secondary nodes and avoid the need for continuous password authentication every time when establishing a connection between nodes, you can configure SSH on each machine to use password-free public key authentication. The configuration is as follows: Run the ssh-keygen command on centos1 to generate public and private keys. Send the public key to the remote machine: ssh-copy-idroot@centos1/centos2/centos3.

(iv) Installing and configuring Hadoop

Configure JDK in the hadoop-env.sh file:exportJAVA_HOME=/java/jdk1.8.0_112
Configure the core-site. XML file, including the Hadoop cluster name, Hadoop file storage and the storage address and ports of hosts in the Zookeeper cluster.

```
< configuration >
  < property >
    < name > fs.defaultFS < / name >
    < value > hdfs : / / ns < / value >
  < / property >
  < property >
    < name > hadoop.tmp.dir < / name >
    < value > / home/ software/ hadoop-2.7.1/ tmp < / value >
  < / property >
< property >
  < name > ha .zookeeper .quorum < / name >
  < value > centos1: 2181, centos 2 : 2181, centos 3 : 2181 < / value >
< / property >
< / configuration >
```

Configure the HDFS -site. XML file. The configuration content is mainly about the number of copies on a cloth cluster, the number of replicas must be 3.

```
< configuration >
  < property >
    < name > dfs.replication < / name >
    < value >3 < / value >
  < / property >
< / configuration >
```

Configure mapred-site. XML to set yarn resource scheduling, and content is as follows:

```
< configuration >
< property >
  < name > mapreduce.framework .name < / name >
  < value > yarn < / value >
< / property >
< / configuration >
```

Configure yarn-site. XML to specify the START VM of Yarn Resource manager, and in this paper it is centos1 and centos3.

```
< property >
< name > yarn .resourcemanager .hostname.rm1 < / name >
< value > centos1 < / value >
< / property >
< property >
< name > yarn .resourcemanager .hostname.rm 2 < / name >?
< value > centos 3 < / value >
< / property >
```

Procedure 1:

Copy the uploaded Hadoop installation package to the/ usr/local directory on the Master node of the cluster, decompress the package, and configure Hadoop related files. The distributed system configuration files are core site. XML files are used to configure the parallel computing model. The configuration procedure is as follows:

So far, you have configured the Hadoop environment in centos1 on the master node. Remotely copy all Hadoop configuration files and installation packages configured on Centos1 on the active node to the same directories on centos2 and centoS3.

Start Zookeeper, Journalnode, Namenode of the first and third VMS, Datanode of the three VMS, Yarn of the first VM, and Resourcemanager of the third VM in sequence. If both the second machine and the third machine can run normally, it indicates that the Hadoop cluster environment is successfully deployed [22]. The distribution of all components in the experimental environment in this paper is shown in Figure 2.

Although the Hadoop cluster has been built up to provide a platform for data storage and computation processing, the hadoop-Eclipse-plugin.jar plug-in needs to be installed in order to develop the calculation code and run the algorithm research program. The plug-in is used to connect to the Hadoop cluster through Eclipse and view and calculate files through the API provided by the Apache official website.

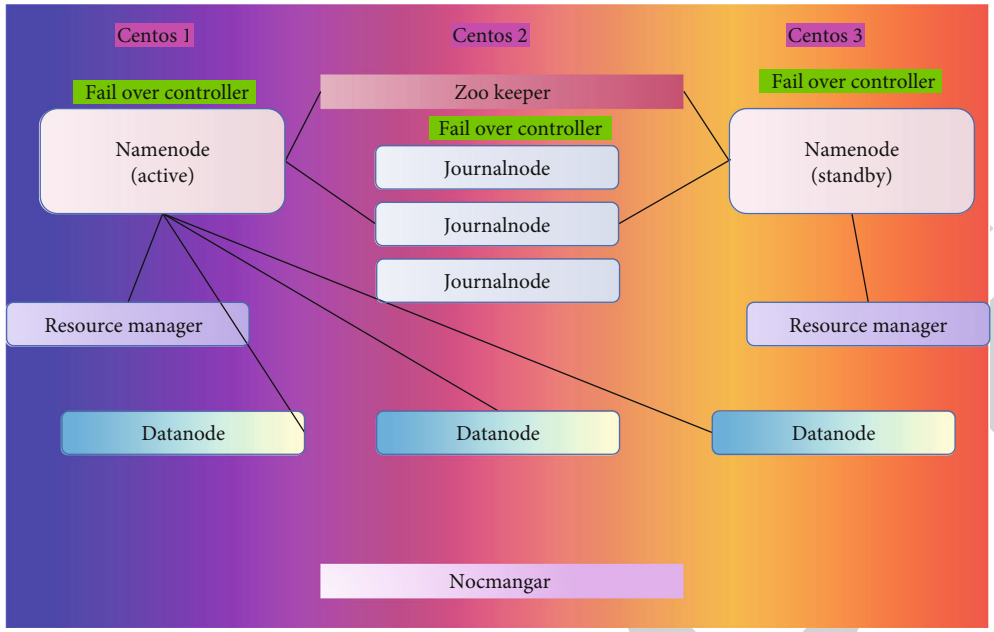

Figure 2: Hadoop cluster architecture diagram.

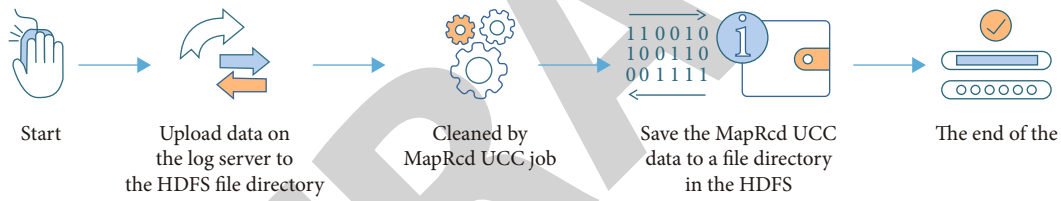

Figure 3: Flow chart of complete repeated data processing.

```
Algorithm: LogMapper
Input:          K, v key-value pairs
Output:         k, v key-value pairs
1               Public class Mapper extends LogMapper<LongWritable, Text, Text, LongWritable>{
2               protected void map(Long Writable key, Text value, Mapper<LongWritable, Text,Text, LongWritable> Context con-
text) throws IOException, Interrupted Excetion{
3               String line -value.to String(;
4               String0 data=line.split(" ");
5               for(String recordodata){
6               context.writ(new Text(record), new Long Writable(1)); }}}
```

Algorithm 1: Pseudocode for the map class.

To configure the plug-in, place the plug-in JAR package in the Eclipse installation directory and decompress it. Restart Eclipse, click the Map/Reduce TAB in Windows, and enter the IP address of the Namenode node: 192.168.175.129 and port: 50020 in the dialog box.

4.3. Cleaning Idea Based on MapReduce Parallelization

4.3.1. Storage of Data. HDFS is used to store the original data and the data after cleaning. HDFS is very friendly for storing and managing large files. According to the size of the data, calculate the number of blocks to be cut, the number of copies, and the Datanodes to store the data through its own Namenode. Then, the HDFS client blocks data to Datanode. To view data files stored in the HDFS, you can use the HDFS API or run the hadoopfs-cat command in Linux.

4.3.2. Calculation of Data. The MapReduce computing model can use APIS to specify paths for reading files, so that any files stored in a distributed system can be read and processed in parallel in a cluster. The processed data is output to a specified file through an OutputStream specified output

```
Algorithm: LogMapper
Input:        k, v key-value pairs
Output:       k, v key-value pairs
1             Public class Mapper extends LogReducer<Text,LongWritable, Text, Text>{
2             @ Override
3             Protected void reduce(Text key, Itcrable<Long Writable> values, Reducer<Text,
                Long Writable, Text, Text> Context context) throws IOExcepttion,
                InterruptedException{
4             String result ="";
5             Int sum=0;
6             for(LongWritable value:values){
7             sum = sum + value.get() +",";}
8             conte: xt.write(key, ncw Text(result));}}
```

Algorithm 2: Reduce class pseudocode.

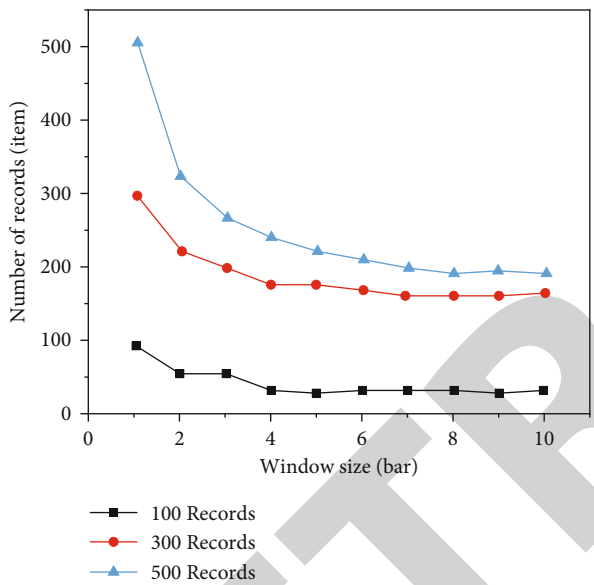

Figure 4: Cleaning results of different window sizes.

path. The working mechanism of MapReduce abstracts the entire process into two parts, and the abstract map and reduce interfaces can implement various algorithms. In this paper, for the cleaning module of completely repeated data, the parallel processing characteristics of MapReduce and the written processing rules are utilized to efficiently calculate completely repeated data [23]. The data storage and calculation process are shown in Figure 3.

4.4. Cleaning Design Based on MapReduce Parallelization

4.4.1. Fully Repeated Data Cleaning Implementation

(1) The Map Class Implements. The map class needs to implement the map method in the mapper interface. The map class inputs the file in the LogDriver class. After MapReduce is processed, the input parameters of the map method are changed to key and value formats. The key indicates the byte offset of each line of data in the CSV file, and the value indicates each line of data in the CSV file. The map method outputs keys and values as well. The pseudocode of map class is shown in Algorithm 1.

(2) Reduce Class Implements. The number and type of the input parameters of the reduce method in the reduce interface must be consistent with the output parameters of the map method. The value here is an iterator, characterized by a collection of values with the same key. Traverse the iterator, get all belong to the same key value, and then, get the value or the value according to their own needs through other processing results. In the reduce method, at this stage in this paper, key is each record, value is the frequency of occurrence of each record, and the number of occurrence of each record is calculated by adding the values of values corresponding to each key. The pseudocode for the reduce phase is shown in Algorithm 2.

(3) Driver Class Implements. The startup class is mainly used to set the map and reduce classes for this processing and the corresponding data output type. In addition, set the path of processing files: to get path and store path. In this document, the original log data is stored in the SY path of vm CENtos1 whose IP address is 192.168.175.129. The final data is output to the theme_click_log_result file in the SY directory of VM CENtos2 whose IP address is 192.168.175.130.

4.5. Analysis of Experimental Results. In this stage, three comparative experiments are conducted. The first group of comparison experiments used SNM algorithm to clean log data and then compared the number of data items in the processed data set to select the value of window size of the traditional SNM algorithm that is relatively more suitable for processing log data. The results are shown in Figures 4 and 5.

In the figure above, 10 window sizes are set for experiments on the basis of three sample data. 100, 300, and 500 samples were selected for verification. The window size is set to 1 to 10, and the step size is 1. The line chart shows the number of remaining data items after data cleaning. For the same data set, the more the number of data items after cleaning is reduced than that before cleaning, the greater the downward trend of the curve in the line graph,

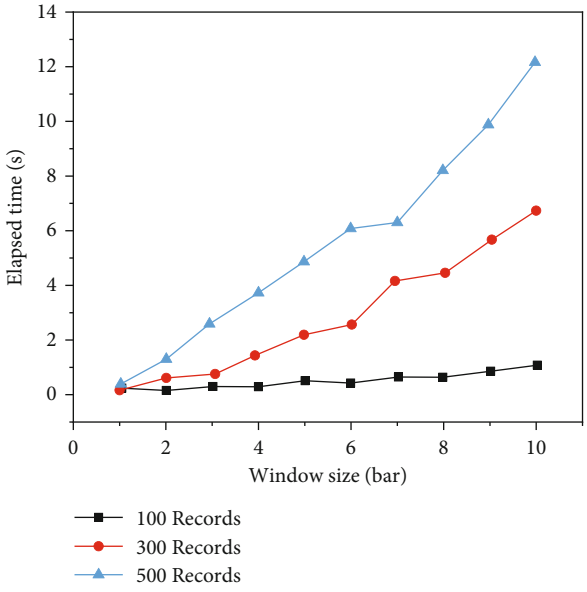

Figure 5: Elapsed time of different window sizes.

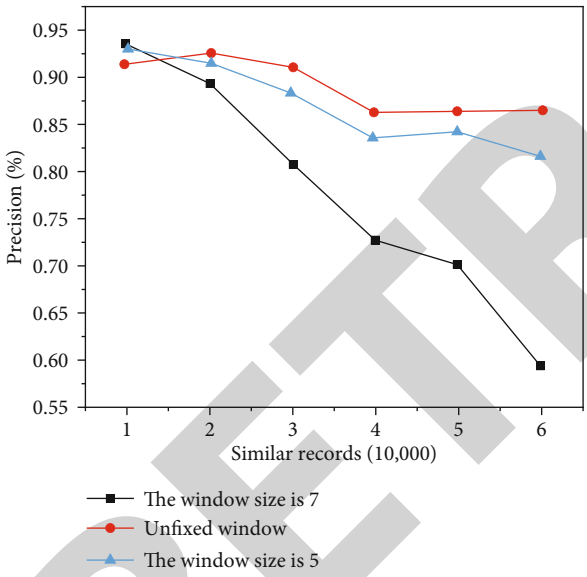

Figure 6: Comparison of accuracy.

indicating that the number of data items cleaned out is more. As can be seen from the figure, for the same data sample, the larger the sliding window is, the fewer remaining records are cleaned; that is, the more data items are cleaned. The downward trend of broken lines all began to decline rapidly, while the cleaning effect of most window sizes did not change significantly in the following years, which was the case for all three sample data. It is not confirmed that the larger the size of the sliding window, the better the cleaning efficiency.

The three broken lines in the figure above represent the data cleaning time under different window sizes. It is not difficult to see that the length of time is positively correlated with the number of records and the size of the sliding window. According to the above experimental results, for all sample data, when the window size is equal to 5, the cleaning effect is not different from that of the window larger than 5. Taking the time factor into account, the window size of 5 is about equal to the median. Therefore, 5 is selected as the initial window size for subsequent experiments [24]. Based on the first group of experiments, the second group of comparative experiments is to compare processing and results of the selection of appropriate window size of the traditional neighbor sorting algorithm and improved nonfixed window neighbor sorting algorithm. It mainly compares the precision rate and time consuming of data cleaning by the traditional algorithm and the improved algorithm. The results are shown in Figures 6 and 7.

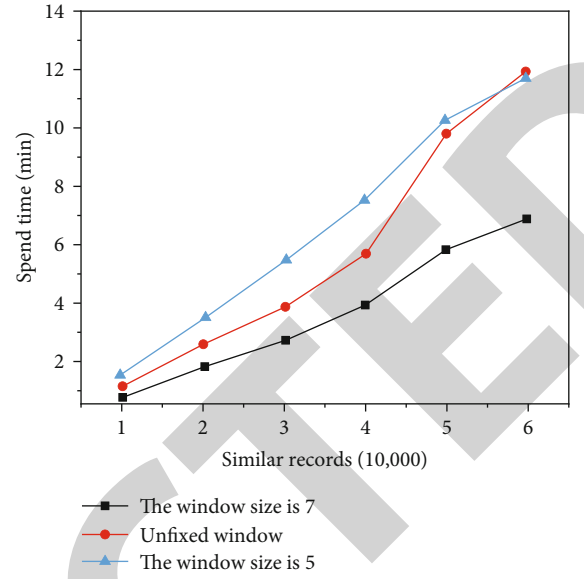

Figure 7: Time comparison diagram.

It can be seen from the figure that the cleaning precision of the traditional neighbor sorting algorithm with window size of 5 and the neighbor sorting algorithm with nonfixed window is significantly higher than that of the traditional neighbor sorting algorithm with window size of 7. Moreover, with the increase of data volume, the accuracy of the algorithm is higher than that of the traditional neighbor sorting algorithm with window size 5.

It can be seen from the figure that the traditional neighbor sorting algorithm with window size of 5 takes the least time to process the same amount of data. The nearest neighbor sorting algorithm with window size 7 is always the longest. The time consumption of the nonfixed window neighbor sorting algorithm is similar to that of the traditional neighbor sorting algorithm with a window size of 5. However, with the increase of data volume, the consumption time increases rapidly until it approaches the consumption time of the traditional sorting nearest neighbor algorithm with window size of 7. Therefore, the algorithm can improve the precision of data cleaning at the expense of cleaning speed.

5. Conclusion

With the rapid increase of data volume in the era of big data, data problems have attracted the attention of the industry. In order to meet the requirements of the times, this paper takes the website log data as the processing object and conducts data cleaning research on the repeated data in the log data. Therefore, this paper combines the research status of repeated data cleaning at home and abroad and adopts appropriate cleaning methods for repeated data processing. For completely repeated data and similar repeated data, Hadoop cluster-related technology and data cleaning algorithm are mainly used to clean log data. The first step of similar duplicate data cleaning is the detection of similar duplicate data. A character frequency-based editing distance (CFLD) algorithm is proposed to detect similar repeated data. Based on the traditional editing distance, the algorithm considers the cost of adding, deleting and replacing and the importance of characters. Character importance is measured by the frequency of occurrence of a character. Characters that appear less frequently can be regarded as more important characters, which reduce the operation cost of important characters, and make the editing distance smaller. In calculating the records with the same traditional edit distance, the records containing important characters are matched preferentially, so as to improve the accuracy of algorithm detection. In this paper, the comparison between the traditional edit distance algorithm and CFLD algorithm shows that the accuracy of the latter algorithm reaches 80.4%, which improves 3.2% on the basis of the edit distance algorithm. It is verified that the improved algorithm can improve the accuracy of the detection.

Data Availability

The labeled data set used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

Acknowledgments

This work is supported by the Shijiazhuang Posts and Telecommunications Technical College.

References

[1] W. Huang, J. Ren, T. Yang, and Y. Huang, "Retracted article: research on urban modern architectural art based on artificial intelligence and gis image recognition system," *Arabian Journal of Geosciences*, vol. 14, no. 10, pp. 1–13, 2021.

[2] L. Feng, J. Wang, C. Ding, Y. Chen, and T. Xie, "Research on the feedback system of face recognition based on artificial intelligence applied to intelligent chip," *Journal of Physics: Conference Series*, vol. 1744, no. 3, article 032162, 2021.

[3] Z. Yang, S. Zhang, R. Li, C. Li, and M. Zhang, "Efficient resource-aware convolutional neural architecture search for edge computing with pareto-bayesian optimization," *Sensors*, vol. 21, no. 2, p. 444, 2021.

[4] H. D. Wu and L. Han, "A novel reasoning model for credit investigation system based on fuzzy bayesian network," *Procedia Computer Science*, vol. 183, no. 19, pp. 281–287, 2021.

[5] Y. Zhou, X. Sun, C. Luo, Z. J. Zha, and W. Zeng, "Posterior-guided neural architecture search," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 6973–6980, 2020.

[6] L. Wu, "Student model construction of intelligent teaching system based on bayesian network," *Personal and Ubiquitous Computing*, vol. 24, no. 3, pp. 419–428, 2020.

[7] M. Scanagatta, A. Salmerón, and F. Stella, "A survey on bayesian network structure learning from data," *Progress in Artificial Intelligence*, vol. 8, no. 4, pp. 425–439, 2019.

[8] G. Costa and R. Ortale, "Integrating overlapping community discovery and role analysis: bayesian probabilistic generative modeling and mean-field variational inference," *Engineering Applications of Artificial Intelligence*, vol. 89, article 103437, 2020.

[9] Y. Iwakami, H. Takuma, and M. Iwashita, "Properly initialized bayesian network for decision making leveraging random forest," *Artificial Intelligence Research*, vol. 9, no. 1, p. 36, 2020.

[10] L. Peng, "An effective analysis of online education model based on artificial intelligence," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 2, pp. 1–12, 2020.

[11] L. Ercanli, "Artificial intelligence with deep learning algorithms to model relationships between total tree height and diameter at breast height," *Forest Systems*, vol. 29, no. 2, p. e013, 2020.

[12] F. Liu, T. Zhang, C. Zheng et al., "An intelligent multi-view active learning method based on a double-branch network," *Entropy*, vol. 22, no. 8, p. 901, 2020.

[13] I. Vendrov, T. Lu, Q. Huang, and C. Boutilier, "Gradient-based optimization for bayesian preference elicitation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 6, pp. 10292–10301, 2020.

[14] F. J. Costello, C. Kim, C. M. Kang, and K. C. Lee, "Identifying high-risk factors of depression in middle-aged persons with a novel sons and spouses bayesian network model," *Healthcare*, vol. 8, no. 4, p. 562, 2020.

[15] M. L. How and L. Wei, "Educational stakeholders' independent evaluation of an artificial intelligence-enabled adaptive learning system using bayesian network predictive simulations," *Education Sciences*, vol. 9, no. 2, p. 110, 2019.

[16] J. Yao and J. Liu, "Research on computer network technology system based on artificial intelligence technology," *Journal of Physics: Conference Series*, vol. 1802, no. 4, article 042028, 2021.

[17] W. Huang and H. Zhang, "Research on artificial intelligence machine learning character recognition method based on feature fusion," *Journal of Physics: Conference Series*, vol. 1544, no. 1, article 012163, 2020.

[18] H. Chen, "Research on innovation and entrepreneurship based on artificial intelligence system and neural network algorithm," *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 2, pp. 2517–2528, 2021.

[19] Q. Liu and Z. Huang, "Research on intelligent prevention and control of covid-19 in china's urban rail transit based on artificial intelligence and big data," *Journal of Intelligent and Fuzzy Systems*, vol. 39, no. 6, pp. 9085–9090, 2020.

[20] Q. Wang and P. Lu, "Research on application of artificial intelligence in computer network technology," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 5, article 1959015, 2019.

[21] Y. Zhang and Y. Pei, "Research on the application of artificial intelligence technology in island tourism," *Journal of Physics: Conference Series*, vol. 1852, no. 3, article 032015, 2021.

[22] J. Li and T. Wang, "Research on the application of artificial intelligence technology in intelligent operation and maintenance of industrial equipment and system," *Journal of Physics Conference Series*, vol. 1992, no. 3, article 032090, 2021.

[23] H. Xiong, "Research on face recognition algorithm based on convolutional nerve," *Journal of Physics: Conference Series*, vol. 1966, no. 1, article 012027, 2021.

[24] L. Huang and G. Liu, "Functional motion detection based on artificial intelligence," *The Journal of Supercomputing*, vol. 78, no. 3, pp. 4290–4329, 2022.