J. COMBIN. MATH. COMBIN. COMPUT. 127b (2025) 6349--6372

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Design of Student Behavior Prediction and Management Strategies Supported by Cluster Computing

Shujuan Li[1,✉]

[1] *Yantai Vocational College, Yantai, Shandong, 264000, China*

ABSTRACT

In recent years, the deepening of reform and opening up, the deepening of the socialization of college management, the trend of students' thinking is more and more diversified leading to the frequent occurrence of college students' behavior. This paper is based on Spark's parallel H-mine cluster computing to mine the behavioral characteristics data of students in frequent item sets. Using the K-Means clustering algorithm optimized by information entropy and density, the clustering and classification process is carried out according to the central value of the obtained behavioral features. Construct the class model of student behavioral features, realize student behavior prediction by K-nearest neighbor algorithm, and build the early warning model of student behavior prediction based on Spark cluster. The results of clustering analysis show that the average number of times a class of students, the second class of students, and the third class of students eat at breakfast is 120.07, 107.66, and 118.25, respectively, and the first class of students has the most number of times of breakfast meals, which shows that this class of students has better eating habits.The number of students studying on March 24, 2023 is predicted by the model based on the K nearest neighbor algorithm, and the trajectory of the real value and the predicted value The number of students with relative error less than 0.2 accounted for 86.42%, indicating that the model is good at predicting the number of students as a whole.

Keywords: Parallel H-mine cluster computing, Density optimization, K-Means clustering, K-nearest neighbor algorithm, Student behavior

1. Introduction

In modern college management, the management and prediction of students' behaviors become more and more important. The daily behavior of college students not only includes classroom learning, interpersonal relationships, cultural and sports activities, but also involves many aspects such as campus safety, online games and social practices [1-3]. Due to the wide range of aspects

✉ Corresponding author.
E-mail address: qimiaosisiruyi@163.com (S. Li).
Received 12 January 2024; Revised 05 March 2024; Accepted 25 December 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-348

involved in behavior, data collection is extremely difficult, and a single data source often fails to comprehensively reflect students' behavioral patterns [4-5]. Traditional management methods rely on manual experience and are difficult to cope with the complex and changing campus environment [6]. Therefore, how to effectively integrate multi-source data and utilize intelligent technology for the management and prediction of student behavior has become an urgent problem [7-8].

With the rapid development of information technology, big data has become an important driving force to promote the innovative development of modern education. By analyzing and processing a large amount of data, educators are able to understand students' learning behaviors more deeply, so as to optimize the teaching methods and learning environment [9-11]. Student behavior prediction model in big data environment has become the focus of research in the field of education. Literature [12] constructed a multi-indicator abnormal behavior prediction method (ABPM-IMISBTI) based on students' behavior and comment text data, integrated objective and subjective labels of students' behaviors through the K-means algorithm, and introduced a long and short-term memory network for behavior prediction. Literature [13] collects students' academic life data from the school's information management and other intelligent departments, and then extracts valuable information about students' behaviors as a way to construct a reliable student behavior prediction model. Literature [14] fully exploits students' behavioral characteristics in consumption, life, and learning based on campus big data, and builds a feature hierarchical model to predict and feedback students' behaviors in a timely manner. Literature [15] studied the student behavior prediction and management methods in the work of academic affairs, and proposed a student behavior prediction model based on convolutional neural network, which can achieve accurate prediction of student behavior by learning the intrinsic characteristics of student behavior data. Literature [16] uses research tools such as data mining to collect rich data specific to campus network users and builds a behavioral analysis and decision-making system based on it, which plays an important role in the exploration of specific patterns and rules of student behavior. Literature [17] developed a student behavior data analysis and early warning management system based on the Hadoop open source platform to form an analytical prediction model and a visualization model of student behavior based on student data in the process of education management, which promotes the digital construction of the campus. Literature [18] points out that the school learning management system lacks an automated intelligent component to analyze the collected student behavior data, and by introducing a machine learning model for training, it can effectively identify and predict learner behaviors and classify each student according to its characteristics, which facilitates the smooth implementation of educational management activities. Literature [19] established an analysis and prediction platform based on students' behavioral characteristics, using data mining algorithms to collect students' consumption, life, study and Internet access data from school management systems, which can effectively analyze students' behavioral characteristics and patterns. However, although the above method strengthens the generalization ability and prediction effect of the student behavior analysis process, it is only applicable to the stage-by-stage prediction and management of student behavior, and cannot satisfy the demand for monitoring the students' learning status during the teacher's lectures.

In recent years, video surveillance systems have been widely used. The school departments concerned spend a lot of manpower and material resources on real-time monitoring of the surveillance videos in the classrooms every year, and by arranging professional teaching staff to watch the collected surveillance videos, they make a unified evaluation of students' behavioral conditions in the classroom [20-22]. The prediction and management of student behavior in the classroom achieved through video monitoring system has become one of the important measures to improve classroom efficiency in scholars' research. Literature [23] shows that the application of visual sensors and computer vision methods in education enhances the ability to automatically detect classroom students' behavioral and emotional states, providing timely feedback for

instructional decisions in the classroom. Literature [24] proposed a multimodal intelligent perception framework fusing video and audio modalities for classroom student behavior recognition, which gained improvement in behavioral prediction progress compared to unimodal approaches. Literature [25] emphasized that accurately and efficiently assessing classroom student behavior is essential to promote the development of high quality teaching practices, and by combining multi-target detection techniques and intelligent algorithms to achieve real-time recognition of classroom behaviors in classroom teaching scenarios with multi-student targets, it can help teachers to understand the students' learning in real time. This technology can predict student behavior to a certain extent, play the reliability of the monitoring system, improve the feedback in real time, not only save a lot of human resources, but also provide a wide range of data information for the statistics of abnormal behavior.

In order to be able to monitor students' violations in the classroom in real time and provide timely feedback, all the surveillance videos are utilized to efficiently analyze and count the students' classroom behaviors, saving the resources of staff scheduling and monitoring quality in order to improve the efficiency of the whole project [26-29]. Thus cluster technology gradually receives attention and develops rapidly. Compared with supercomputers, clusters are able to add and remove nodes arbitrarily for better scalability, and the hardware equipment is much cheaper than supercomputers, but provide comparable performance [30-32]. Due to the need for real-time transcoding of a large number of videos, which involves a large amount of computation, the use of clusters can solve the problem of insufficient computational power, and the use of clusters instead of a single machine to complete complex work tasks has become a trend [33-35].

In this paper, we utilize the F-List parallel method to transform the F-list into the HStruct data structure that meets the requirements of the H-mine algorithm, and parallelize the frequent itemsets. The entropy method is used to calculate the feature weights of each attribute to complete a more refined calculation of the Euclidean distance between data points. The standard deviation is adopted as the standard measurement function to obtain the objective value function of the empowered category, and the clustering contribution played by each attribute in the clustering process of the data object is analyzed using the K-Means clustering algorithm. Based on the obtained initial feature set, construct the initial student behavior feature set, predict the probability of occurrence of student behavior using the K-nearest neighbor algorithm, and run the student behavior prediction model in the Spark cluster to achieve student behavior prediction and warning. Aiming at the results of student behavior prediction, the student behavior management strategy is designed.

2. Prediction of Student Behavior under Cluster Computing

2.1. Spark-based computational model for parallel H-mine clusters

2.1.1. Input data organization. In order for Spark to process the data correctly, it is assumed that the data format is <line content, line number>, each data item in the line content is separated by a space, the separator between the line content and the line number is the same as that of the data items in the line content (e.g., <dsaf2>), in which the line content is <ds af> and the line number is 2. The reason for adding the line number at the end of each line is threefold: first, the line number can be used as a separator between lines without changing the relative position of the elements between lines, making the lines orderly. In the premise of not changing the relative position of the elements in the line, so that the line is organized, the second is the use of line number as a line and line separator, the third is the statistics of frequent 1 item set, as long as the number of support is greater than 1, you can filter out the line number, does not affect the normal operation of the program. The support number is greater than 1 can filter the line number, the proof is as follows: because the line number value is unique, so for the dataset without line number, N realizes the parallel H-mine algorithm is divided into the F-list parallel method, the F-list transformed into the HStruct parallel method, the

load balancing method, and the parallel mining of the frequent itemset method in four steps. -N a line number value appears only once in the data set and at the end of each line, so as long as the support number is greater than 1, that is, the support is greater than (N is the total number of lines) can be filtered line number. In summary, when the support is greater than one, the row number does not affect the frequent itemset mining.

2.1.2. Parallel H-mine Algorithm:

1) F-list parallel method. Frequent item projection is the key to mining frequent itemsets in H-mine algorithm, which means that under the premise of constant relative positions of elements between rows, all elements within rows are frequent items (i.e., data items in F-list that satisfy the minimum support), and the elements within rows are ordered.F-list, i.e., frequent 1-itemsets, is a necessary condition to realize the frequent item projection, and in the mining of frequent itemsets based on the H-mine algorithm Prior to the H-mine algorithm-based frequent itemset mining, the F-list needs to be transformed into an HStruct data structure that satisfies the requirements of the H-mine algorithm. Therefore, in integrating H-mine into Spark environment, it is necessary to realize parallel processing of F-list.

2) F-list into HStruct parallel method. The F-list into HStruct parallel generation method is divided into two phases, the first phase obtains frequent item projections in parallel, and the second phase uses the results of frequent item projections and F-list to transform the F-list into HStruct, which prepares for the subsequent realization of parallel frequent item set mining. These parallel processes can reduce the algorithm preprocessing time and thus can reduce the overall running time of the algorithm. Before transforming F-list into HStruct, frequent term projection needs to be obtained. It refers to filtering out the data items that satisfy the minimum support in each row and sorting the frequent items in each row in ascending order in terms of rows, which facilitates the subsequent transformation of F-list into HStruct.

3) Load balancing method. After transforming F-list into HStruct (hereinafter referred to as FlistHStruct), frequent item sets can be mined. From the frequent items <A, B, C, E> in FlistHStruct, the frequent item sets generated with the help of frequent item projection can be categorized into four types, and these four types of frequent item mining processes are independent of each other. Therefore, in parallel processing, these four types of frequent itemset mining can be divided into up to four partitions simultaneously. Without considering I/O blocking, assuming that the time used to compute each frequent item is equal, in order to make the running time on each partition approximately equal, it is equivalent to find the total number of frequent items computed by each partition is equal.

Assume that there is n data item $\{x_1, x_2, \ldots\ldots x_n\}$ in FlistHStruct, and that in the best case there are 2^{n-k} frequent items prefixed with $x_k (1 \le k \le n)$. The proof is as follows:

In the best case, both x_k and the $t(t > k)$ th term can be combined and x_k is always in the first place. Using the unordered nature of the set, for a frequent $(n-k+1)$ item set, the next $(n-k)$ data items after x_k are stored in an order-independent position, e.g., $\{x_k x_{k+1} x_{k+2} \ldots\ldots x_n\}$ and $\{x_k x_{k+2} x_{k+1} \ldots\ldots x_n\}$ are the same frequent $(n-k+1)$ item set. x_k as the prefix of the frequent items of a total of $(n-k+1)$ kinds of sets, divided into frequent 1 sets, frequent 2 sets ..... frequent $(n-k+1)$ sets, statistics for each kind of frequent items set contains frequent items of the number of items, and then add up, you can find out to x_k as the prefix of the total number of frequent items, that is:

$$\sum_{x=1}^{n-k+1} (\text{Frequent } x \text{ item set}) = C_{n-k}^{0} + C_{n-k}^{1} + \ldots\ldots + C_{n-k}^{n-k} \tag{1}$$

by the binomial theorem:

$$\sum_{r=0}^{n} C_n^r a^{n-r} b^r = (a+b)^n \tag{2}$$

When $a=1$ and $b=1$, get:

$$C_n^0 + C_n^1 + \ldots\ldots + C_n^n = 2^n \tag{3}$$

Equations (1), (2) and (3) are obtained by joining them:

$$\sum_{x=1}^{n-k+1} (\text{Frequent } x \text{ item set}) = 2^{n-k} \tag{4}$$

From equation (4), there are a total of $2^{n-k}(1 \le k \le n)$ frequent terms prefixed with x_k in the best case.

Clearly, when there are n data items in FlistHStruct, a total of $2^n - 1$ frequent terms need to be computed in the best case, i.e., $\sum 2^{n-k} = 2^n - 1 (1 \le k \le n)$.

The H-mine algorithm mines all frequent terms of one partition at a time on a single machine, and for parallel mining, frequent terms of multiple partitions can be mined simultaneously [36]. Although the partitions are parallel to each other, the mining is serial for the same partition. Therefore, it is crucial to achieve load balancing by ensuring that the total number of frequent terms mined in each partition is approximately equal without considering issues such as computer I/O blocking, available memory, etc., i.e., in the ideal case [37].

From Eq. $x_k = 2^{n-k}(1 \le k \le n)$, it can be seen that $x_1 = 1 + \sum x_k (2 \le k \le n)$. If it is divided into 2 partitions, ideally, the 1st partition can only compute all the frequent items prefixed with x_i, and the 2nd partition computes the remaining frequent items. However, 2 partitions cannot fully utilize the resources of the computer. Studies have shown that the number of partitions is generally chosen to be 2~3 times the number of computer cores. In this paper, according to the experimental environment, the number of partitions in the approximate ideal case is set to 4. When the number of partitions is 4, according to the current method, the 1st partition can only compute all the frequent items prefixed by x_1, the 2nd partition can only compute all the frequent items prefixed by x_2, the 3rd partition can only compute all the frequent items prefixed by x_3, and the last partition computes the remaining frequent items. However, there are significant problems with this method.

As can be seen from $x_k = 2^{n-k}(1 \le k \le n)$, $x_{k-1} = 2x_k (k \ge 2)$. Ideally, the total number of frequent terms mined by the 1st partition is twice that of the 2nd partition, the total number of frequent terms mined by the 2nd partition is twice that of the 3rd partition, and the total number of frequent terms mined by the 3rd and 4th partitions are approximately equal. Assuming that the time to mine each frequent term is equal, it is clear from the above analysis that the total time used to mine the partition with the most frequent terms is four times the total time used to mine the partition with the least frequent terms. Therefore, there is a serious computational skew problem. To solve this problem, FlistHStruct needs to be processed twice.

Firstly, x_i in FlistHStruct is written to a file, then frequent 2-item set mining is performed on x_i, and finally the frequent 2-item set results are combined with all frequent 1-item sets except x_1 to form the new input data. From the above analysis, it can be seen that, ideally, there are n-1 frequent 2-term sets prefixed with x_1 for $x_1 x_k (2 \le k \le n)$. There are $2^{n-k}(2 \le k \le n)$ frequent term sets prefixed with $x_1 x_k (2 \le k \le n)$.

4) Mining frequent term sets in parallel. As can be seen from subsection (3), parallel mining is realized by partitioning the final result of the secondary processing FlistHStruct (hereinafter referred to as SecFlistHStruct) to ensure that the total number of frequent items mined in each partition is approximately equal, and then finally the frequent items mined in each partition are merged to generate the final frequent item set, in which the partitioning method is to, partition the The partitioning method is to divide the first item of SecFlistHStruct frequent 2-item set into the first

partition, then divide the other items of the remaining frequent 2-item set into the second partition, then divide the first item of SecFlistHStruct frequent 1-item set into the third partition, and finally divide the other items of the remaining frequent 1-item set into the last partition. When the total number of items in the SecFlistHStruct frequent 2 item set is large enough, load balancing is achieved to some extent. However, when the total number of SecFlistHStruct frequent 2-item set items is only 1, it may result in some partitions not being able to get data. Therefore, when the total number of SecFlistHStruct frequent 2-item set items is only 1 item, hash partitioning is used. In Spark, custom partitions need to inherit the Partitioner abstract class, and be sure to implement the numPartitions and getPartition methods in it.

2.2. Traditional K-Means Clustering Algorithm

2.2.1. Traditional K-Means Clustering Algorithm Flow. K-Means algorithm is a clustering algorithm for the detection of the division of the problem of differences between the data elements, it is first selected from the sample set of K to start the clustering target, and then according to the rules of the algorithm requirements, the calculation of the distance between the sample data objects to be carried out, according to the results of the calculation of the analysis of the sample data objects grouping, through iterative calculations until the center of the clustering and then no change, to get K clustering results.K-Means K-Means clustering algorithm implementation process is as follows [38]:

Input: set the sample set S has n data objects and the number of clusters K:

$$S = \{x_1, x_2, \cdots x_n\}, K = \{c_1, c_2 \cdots, c_k\} \tag{5}$$

Output: K cluster that satisfies the minimum criteria of the criterion function.

The processing flow is as follows:

1) Arbitrarily select K samples from the sample set S as the initial clustering center.

2) According to the mean value of each clustered sample, reclassify the corresponding object by calculating the distance d between each sample and the center object (Euclidean distance is used to represent it in this paper), and then reclassify the corresponding object according to the calculated minimum distance. Let two p -dimensional sample data points $x_i = (x_{i1}, x_{i2}, \cdots, x_{ip})$ and $x_j = (x_{j1}, x_{j2}, \cdots, x_{jp})$, the Euclidean distance between them is defined as equation (6):

$$d(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots (x_{ip} - x_{jp})^2} \tag{6}$$

The average distance of all sample data points is (7):

$$Meandist(S) = \frac{2}{n(n-1)} \times \sum_{i \neq j, i, j=1}^{n} d(x_i, x_j) \tag{7}$$

3) Recalculate the mean value (center object) of each sample that has been obtained.

4) Cycle through steps (2) and (3) above until the value of the objective function is constant or less than a specified threshold.

The objective function is the squared error criterion function, defined as equation (8):

$$\sigma_i = \sqrt{\frac{\sum_{i=1}^{n_i} (x_i - c_i)^2}{|C_i| - 1}} \tag{8}$$

where c_i is the center-of-mass point of a data object of the same class, and the center-of-mass point c_i is defined in Equation (9):

$$c_i = \frac{1}{|C_i|} \sum_{x_j \in T_i} x_j \tag{9}$$

c_i denotes the center of the i nd cluster and $|C_i|$ is the number of data objects in class C_i.
5) End and obtain K clusters.

2.2.2. K-Means Clustering Based on Information Entropy and Density Optimization. Through the analysis of the clustering contribution played by each attribute in the clustering process of data objects, the entropy value method is used to calculate the weights of each attribute feature to complete a more refined calculation of the Euclidean distance between data points and clustering [39]. The steps for calculating the feature weights using the entropy value method are as follows.

Stepl construct the attribute value matrix as shown in equation (10):

$$A=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \tag{10}$$

Equation (10) in which m represents the number of dimensions of the data object and n represents the number of sample data.

Step2 calculates the weight of each attribute of the data object, and the weight of the attribute value of the i th data object corresponding to the j rd dimension attribute. First of all, the data need to be processed to compress the data density range to the interval [0, 1], the process is shown in the formula (11) type:

$$M_{ij}=a_{ij}/\sum_{i=1}^{n}a_{ij} \tag{11}$$

Equation (11) where a_i represents the attribute value and M_{ij} represents the attribute value weight, where $i=1,2,\cdots,m, j=1,2,\cdots,n$

Step3 calculates the entropy value formula for the j th dimensional attribute as shown in (12):

$$H_j=-\frac{1}{\ln n}\sum_{i=1}^{n}M_{ij}\ln M_{ij} \tag{12}$$

When $M_{ijn}=0$, there is $M_i \ln M_{ij}=0$. If for a given j value a_i are all equal, then:
$M_{ij}=a_{ij}/\sum_{i=1}^{n}a_{ij}=1/n$, at this time the H_j takes an extremely large value.

Step4 Calculate the coefficient of variation for the j th dimensional attribute as shown in equation (13):

$$q_j=1-H_j \tag{13}$$

Where the coefficient of variation is q_j. For a given j, when H_j is smaller and q_j is larger, the attribute is more important When H_j is larger and q_j is smaller, then the attribute's clustering role is smaller. When $H_j=1$, q_j is zero, then the clustering role of the attribute is zero.

Step5 Calculate the attribute weights in dimension j The formula is shown in (14):

$$w_j=q_j/\sum_{j=1}^{m}q_j \tag{14}$$

Step6 Use the Euclidean distance formula to calculate the similarity between data objects, according to the definition can be obtained after the assignment of the Euclidean distance as shown in Equation (15):

$$d_w=(x_i,x_j)=\sqrt{\sum_{p=1}^{m}w_p(x_{ip}-x_{jp})^2} \tag{15}$$

In Equation (15), w_p is the weight value of the attribute in the p nd dimension. It is equivalent to making the attribute values corresponding to the weights appropriately enlarged or reduced, so that attributes with large weights have a greater role in clustering and attributes with small weights have a smaller role in clustering.

Step7 adopts the standard deviation as the standard measurement function to find the assignment category objective value function as shown in Equation (16):

$$\sigma_i = \sqrt{\frac{\sum_{a_i \in T_j} d_w(a_i, c(C_j))}{|C_j| - 1}} \tag{16}$$

In formula (16), σ_i represents the standard deviation of the assignment of class i, and $|C_j|$ is the number of data objects contained in C_j. As can be seen from formula (16), the smaller the value of σ_i, the greater the similarity of the data objects within the class, the denser the data objects, and the more the center of mass of the class where the data objects are located can reflect the classification decision surface.

2.3. Prediction of Student Behavior

2.3.1. Modeling Student Behavioral Characteristics Categories. The initial feature set constructs the initial student behavioral feature set, which contains various elements such as age, gender, and student evaluation indicators:

$$C = \{S_1, S_2, S_3, ..., S_m\} \tag{17}$$

Weights are assigned to each element of the set of student characteristics to accurately measure the degree of contribution of each characteristic in the model, in addition this analysis needs to satisfy as shown in equation (18):

$$\sum_{i=1}^{m} w_i = 1 \tag{18}$$

Student behavioral characteristics based on the various dimensions of the indicators that represent the school behavior at school, and give each feature weight values based on student behavior to build a category model. It is worth stating that the weights are also set for personal information such as the students' faculties, because the faculties are different, and for the same faculty in the same environment, the similarity is higher and the prediction effect is better, so the faculties more or less will also affect the student indicators:

$$C_i = S_i W_i \tag{19}$$

The model developed in this case mainly utilizes the entropy weight method in the method of establishing the weight values. This method refers to the use of statistical methods to process the data and obtain the weight values. The $n*m$ st order matrix $U = (u_{ij})_{nm}$ is used to represent the value of student's behavioral characteristics, where u_{ij} is used in this equation to represent the value of i the j th behavioral characteristic. The characteristic dimension of students is represented by M. The entropy weight method is calculated by combining the following steps:

1) Obtain the initial student behavioral feature matrix:

$$U = \begin{pmatrix} u_{11} & \cdots & u_{n1} \\ \vdots & \ddots & \vdots \\ u_{1m} & \cdots & u_{nm} \end{pmatrix} \tag{20}$$

Where u_{ij} is the value of the i rd evaluated object under the j nd indicator.

2) Calculate the characteristic weight of the i th evaluated object under the j indicators. Remember that the characteristic weight is P_i, then:

$$P_{ij} = \frac{u_{ij}}{\sum_{i=1}^{m} u_{ij}} \tag{21}$$

3) Calculate the entropy value of student behavioral characteristics for item j

2.3.2. Analysis of Student Behavior Prediction Methods. Figure 1 shows the student behavior prediction model based on K nearest neighbor algorithm. First of all, student behavioral characteristics can be obtained through the student evaluation indicators, and based on this to construct the student behavioral category model, based on which the K nearest neighbor algorithm is used to formulate a model to predict student behavior. This prediction of student behavior indicators are students to be expected semester behavior performance prediction, for example, according to the third semester of the senior behavior can be predicted by the predicted student in his third semester behavior performance.

Combined with the personal information of the students on the basis of the category model to determine the student's grade level, from which a sample of the senior students involved in this prediction is extracted and the similarity between the two is calculated to achieve accurate prediction. In this session, a standardized method of calculating the similarity of student behavior was used. And this was used to identify the K seniors in the senior class who are closest to the target student x. Combining the behavioral characteristics of the K neighboring seniors during the prediction semester was used as a basis for predicting the behavior of the junior students, specifically in a non-parametric regression-based prediction of the target. The degree of similarity between two different groups of students in the training sample set and the target students was calculated. In order to more accurately reflect the correlation between the two, this paper introduces a weighted Euclidean distance. It is assumed that the distance indicates the existence of a large distance, which means that the similarity between the two groups is low.

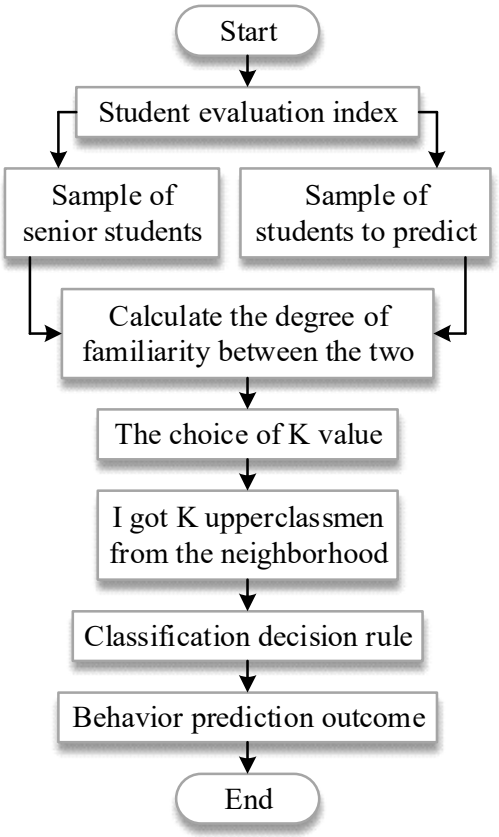

Fig. 1. student behavior prediction model based on K-nearest neighbor algorithm

The specific steps are as follows:

1) Calculation of distance. Assuming that the behavioral characteristics of all students belong to the same n-dimensional space, and the x of all students are represented as feature vectors, i.e., $x_i, x_j \in R^n, x_i = (x_i^{(1)}, x_i^{(2)}, ..., x_i^{(n)})^T, x_j = (x_j^{(1)}, x_j^{(2)}, ..., x_j^{(n)})^T$, R^n is a n-dimensional real vector space, the similarity situation of the two types of students can be calculated based on the equation as in (22):

$$L(x_i, x_j) = \left(\sum_{l=1}^{n} | x_i^{(l)} - x_j^{(l)} |^2 \right)^{\frac{1}{2}} \tag{22}$$

Combining the existing Euclidean distances and introducing the weight values of each feature, the weighted distance formula of Eq. (23) can be obtained and the correlation value can be calculated:

$$d(x_i, x_j) = \left(\sum_{l=1}^{n} |w_i (x_i^{(l)} - x_j^{(l)})|^2 \right)^{\frac{1}{2}} \tag{23}$$

When the distance formula becomes $d(x_i, x_j) = \left(\sum_{l=1}^{n} | w_i (x_i^{(l)} - x_j^{(l)}) |^2 \right)^{\frac{1}{2}}$, the similarity of each attribute mainly depends on the weight value when the distance to the students is calculated, and the science gives the indicators corresponding weights, which is more conducive to improving the accuracy of K neighboring students.

2) Selection of K value. After the calculation of the distance, it is still not possible to determine what the K value is, because the neighboring distance does not know how much is appropriate to take, so it is also impossible to determine the K value. How to choose the K value will affect the results of the

K nearest neighbor method to some extent.

Assuming that the chosen K value is small, that is to say, the prediction of the training example of the domain is small, so it will be a certain degree of "learning" to reduce the approximation error. Objectively speaking, the key to accurate prediction is to input similar training examples. However, this is very easy to appear "learning" estimation error is large, the prediction results will be sensitive to the near-neighboring instances, and then if its instances are noisy, it will inevitably increase the prediction error rate. In other words, reducing the K value is equivalent to complicating the overall model, which is the main cause of overfitting.

Assuming that the chosen K value is large, i.e., it means that the domain of predicting training examples is large, which has the effect of simplifying the estimation error. However, there is the same shortcoming, i.e., the possibility of increasing the approximation error. The fact that its inputs are not similar, i.e., the input instances are farther away can also be detrimental to the accurate prediction of the training instances, and again, it is highly susceptible to prediction errors. In summary if the value is too large, it has the effect of simplifying the overall model.

For the selection of the K value, cross-validation is usually used to select the optimal K value, which is generally lower than the square root of the number of training samples. There are already open-source cross-checking methods, which can be used to realize the K value determination of the K nearest neighbor algorithm by calling the relevant methods.

3) Prediction results are determined by classification decision rules. It is well known that the K nearest neighbor method often exists in the classification decision rule of majority voting, that is, by the majority of the K neighboring students of the students to be predicted class to decide the class of students to be predicted, briefly, is the minority to the majority.

Its specific rule is described below:

Assume that the loss function is a 0-1 loss function and the classification function is:

$$f : R^n \to \{c_1, c_2, ..., c_k\} \tag{24}$$

Then the probability of misclassification is:

$$P(Y \neq f(X)) = 1 - P(Y = f(X)) \tag{25}$$

The K nearest neighboring student samples obtained form set N, where the category is assumed to be c_j. The corresponding misclassification rate can be calculated using the following equation:

$$\frac{1}{k}\sum_{x_i \in N} I(y_i \neq c_j) = 1 - \frac{1}{k}\sum_{x_i \in N} I(y_i = c_j) \tag{26}$$

Minimizing the empirical risk is possible only when $\sum_{x_i \in N} I(y_i = c_j)$ reaches its maximum value, based on which the majority voting rule can be interpreted as minimizing the empirical risk. The final formula is expressed as shown in equation (27).

$$y = argmax \sum_{x_i \in N} I(y_i = c_j), i = 1, 2, ..., N; j = 1, 2, ..., K \tag{27}$$

Where, $x_i \in R^n$ is the feature vector, $y_i \in \gamma = \{c_1, c_2, ..., c_k\}$ is the category of the senior sample, and I is the indicator function i.e. I is 1 when $y_j = c_j$ and 0 otherwise I.

Finally, the obtained value of y is the final prediction result.

2.4. Spark cluster-based implementation of student behavior prediction and warning

2.4.1. Early warning modeling of student behavioral characteristics. Student behavior can be quantified through students' habits in school, life trajectory, learning habits and other behaviors, through the establishment of the student "portrait" feature library, the students in school consumption, learning efforts, regularity of work and rest time, learning skills and other

multi-dimensional attributes of the characteristics of the analysis, so that the school management personnel can carry out personalized and accurate management provides a scientific basis. This can provide a scientific basis for school administrators to carry out personalized and precise management.

The process of building the model of students' behavioral characteristics:

1) Construct a set of student behavioral characteristics, assuming that the set of student behavioral characteristics $S=\{C_1,C_2,\cdots C_m\}$, whose C_i represents the attributes representing the school behavioral characteristics, such as will be the student's gender, grade level, affiliated colleges, card spending, average grades and other characteristics.

2) Different weights are assigned to student behavioral features $\sum_{i=1}^{m} w_i = 1$, where w_i represents the weights assigned to the i th behavioral feature of students.

3) The framework of student behavior prediction and early warning model is shown in Figure 2. Student behavior early warning model building. After extracting the behavioral features from the student portrait feature library, the weights of student behavior indicators are calculated, and then the student classification warning results are obtained by training the warning model.

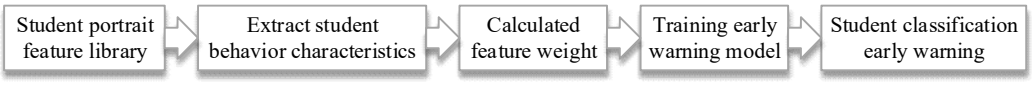

Fig. 2. Framework of student behavior prediction and early warning model

2.4.2. Parallelization based on Spark technology. Parallelization of cluster computation is achieved by using parallel H-mine algorithm, in which Spark adopts Master-Slave architecture, cluster resource management service acts as a master node to control the whole cluster, and the control computation node is responsible for running the job tasks and launching the execution process with specific tasks or task control node for each application.

2.4.3. Model building process:

1) Install the downloaded JDK, Hadoop, Spark installation files.
2) Configure environment variables: create environment variables and assign values, such as SPARK_HOME, SPARK_CLASSPATH, HADOOP_HOME, JAVA_HOME.
3) Add the Hadoop and Spark bin file directories to the path variable.
4) Network configuration: set the IP addresses of Master host and Slave host to 10.1.1.11-10.1.1.14.
5) Configure SSH passwordless login between host nodes.
6) Configure the configuration file for Hadoop.
7) Configure Spark's configuration file.
8) Format HDFS: hadoop namenode-format.
9) Start the HDFS cluster.
10) Start the Spark service and test the installation.
11) Build Spark development environment.

3. Analysis of results in predicting student behavior

3.1. Cluster analysis results

3.1.1. Data analysis techniques. In this paper, 1542 students' behavioral data were collected, their attributes were labeled by creating portraits, and correlation validation and cluster analysis were conducted for students' behaviors, and the dimensions of analysis contained the number of effective operations, online hours, after-school hours, and course hours. Figure 3 shows the results of

clustering analysis of the original sample data, from the comprehensive score, the subject part of the students' scores are between 64 and 90 points, which is highly representative.

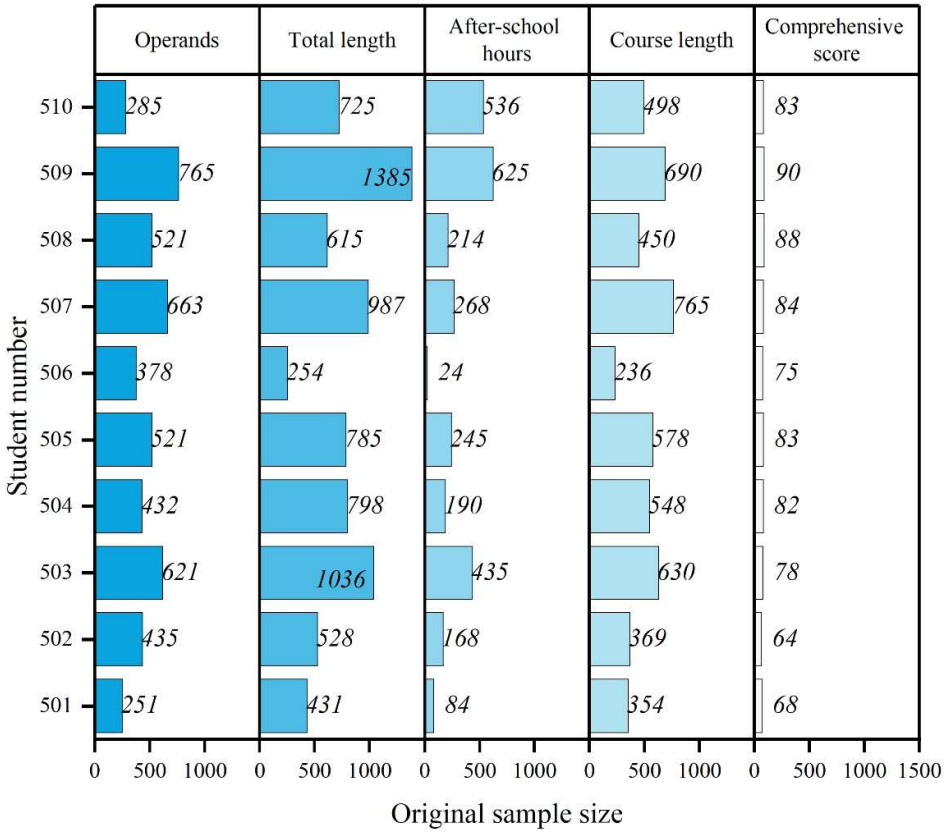

Fig. 3. The cluster analysis is based on the original sample number

3.1.2. Cluster classification. Based on the sample data of the above 3 dimensions which have some correlation with the comprehensive score, the next step is to analyze the K-Means clustering algorithm.The K-Means algorithm settings are compared horizontally to see the classification effect and analyzed. In the process of performing iterations, the sum of error squares is used to adjust the optimal number of iterations. Figure 4 shows the two-dimensional section of the data amplification effect in the case of the optimal iteration. At the same time, the clustering effect of 1542 students combined with the single-point data of students' behaviors for cluster analysis, and it was found that each cluster implies the corresponding behavioral information. For example, students with better final grades are generally characterized by a high number of effective operations and high after-school hours, while students with lower final grades are generally characterized by a low number of effective operations and fewer online hours. The figure further shows the number of 3 categories and behavioral characteristics of the cluster analysis, the number of students in categories 1 to 3 are 403 (26.13%), 712 (46.17%) and 427 (27.7%) people respectively, and the level of the behavioral characteristics represent the strong, average and weak practical skills.

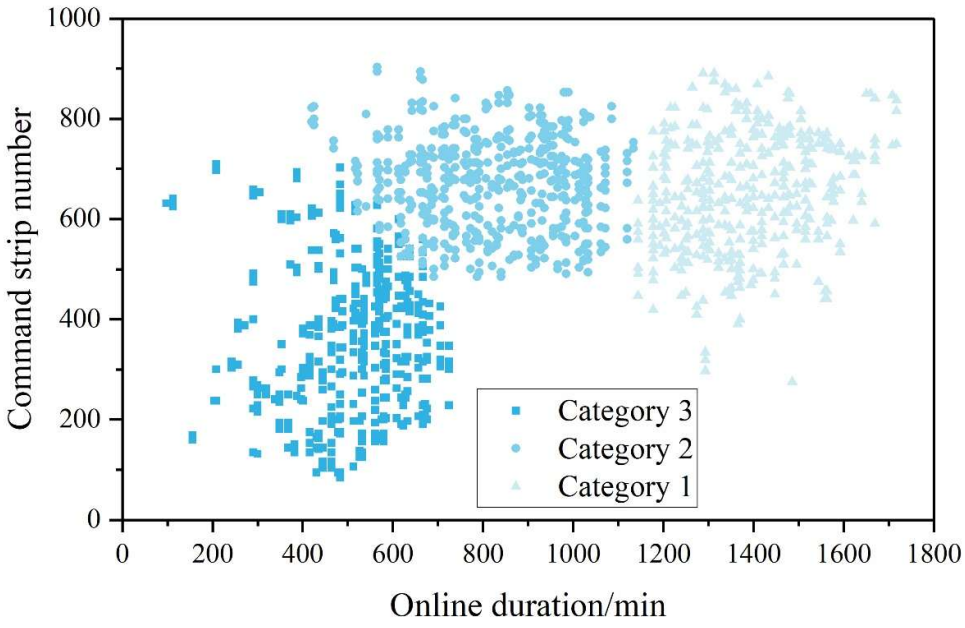

Fig. 4. K-Means (k=3) two-dimensional section

3.2. Analysis of the effect of parallel H-mine algorithm application

3.2.1. Behavioral data. Under the premise that the input data is sparse, it often contains more null values, which is more suitable for the H-mine algorithm to mine the frequent items in the student data. Figure 5 shows the relationship between the degree of support and the number of frequent items, when the H-mine algorithm frequent items set for mining, you need to set the degree of support as well as the maximum length of frequent items of the two parameters. The support degree is the proportion of one or more features in the whole at the same time, and the number of frequent items has a direct correlation with the setting of the support degree. The maximum length of the frequent items is the maximum frequent item length, which is usually set to the number of features, and it is set to 8. The data support is the input data after preprocessing, and the support degree of the experiment is set to 0.01, 0.03, 0.05, 0.08, 0.1, 0.2, 0.3, 0.4, and 0.5, respectively, and the support degrees of the nine different kinds of support degrees are applied in the parallel H -mine algorithm, divided into nine experiments, in the frequent terms of the nine different degrees of support corresponding to the total number of 1235, 492, 276, 155, 104, 35, 13, 8, 1. When the degree of support is greater than 0.2, the frequent terms can not be used in the analysis because of its small number, the minimum degree of support will be set to 0.1, at this time, the frequent term There are 105 frequent terms, so the student behavior is analyzed using the set of frequent terms in this algorithm.

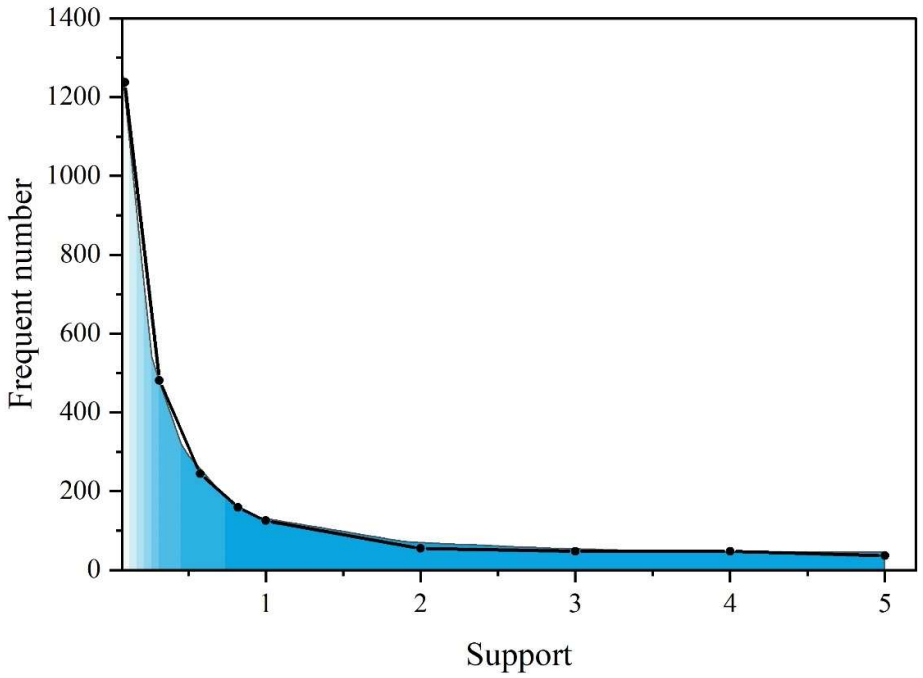

Fig. 5. The relationship between support and frequent Numbers

3.2.2. Dining behavior. The grades of the data of the research subjects were selected for ranking, and according to the clustering results, they were divided into three categories of good, medium and poor, and set the labels as "1", "2" and "3", which accounted for 19.24%, 61.52% and 19.24% of the total number of students, respectively. 19.24%, 61.52%, and 19.24%, respectively, on the basis of which the achievement ranking data and campus behaviors were analyzed, so as to transform the initial behavioral data into behavioral characteristics related to academic performance. In order to avoid excessive data errors, student behavior data were analyzed independently for College One. Figure 6 shows the statistics of three types of students in College 1, and the average number of meals for students in Class 1, Class 2 and Class 3 students was 120.07, 107.66 and 118.25 respectively, and the average number of meals at lunch was 186.77, 194.65, 184.49, the average number of meals at dinner was more than 170, on the whole, the number of breakfast meals for the three types of students was much less than the number of lunch and dinner, which to a certain extent indicated that most of the students had irregular eating, among which, in comparison, the first group of students in the three groups had the most breakfast times, indicating that such students had better eating habits.

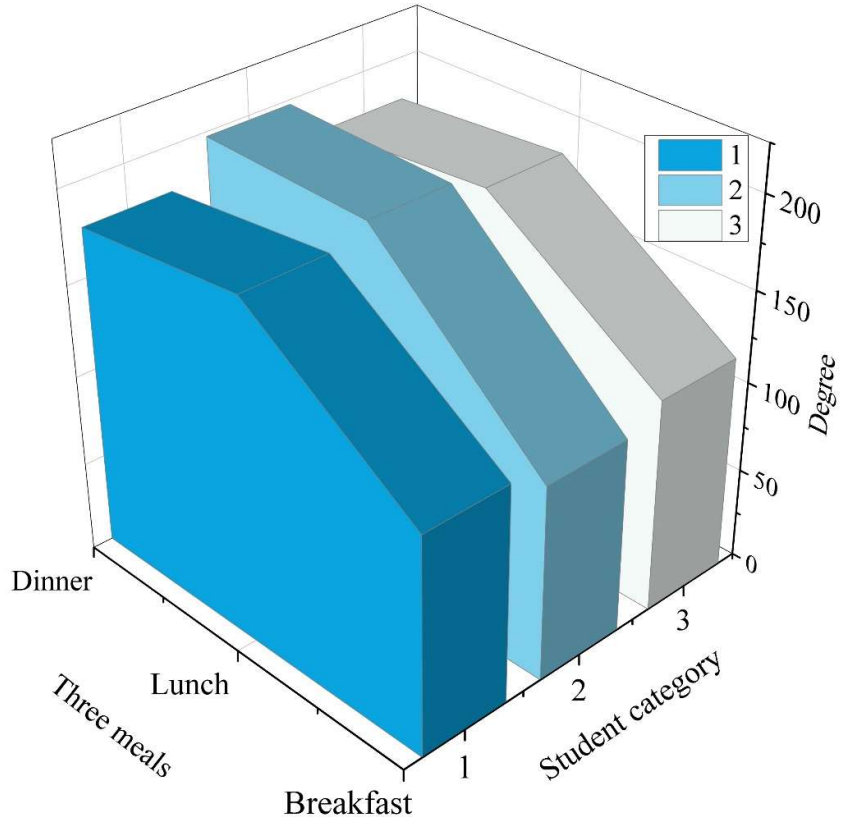

Fig. 6. The three kinds of meals statistics of three students

3.2.3. Borrowing behavior. Figure 7 shows the behavior of the three categories of students during exam and non-exam time periods, as can be seen from the difference in the number of times the three categories of students went to the borrowing of books, the documentation center, and the library during exam and non-exam periods, the average number of times students went to the library was much higher than the average number of times they went to the borrowing of books and to the documentation center, and the average number of times the three categories went to the library was 52.5, 50, and 41.5 times, respectively. The behavior of borrowing books and studying in the library is in a downward trend with the differences in student categories. Specifically, the first category of students have better study and borrowing habits, while the third category of students have relatively fewer study behaviors, indicating a lack of practical action, which allows for the analysis of student behavior.

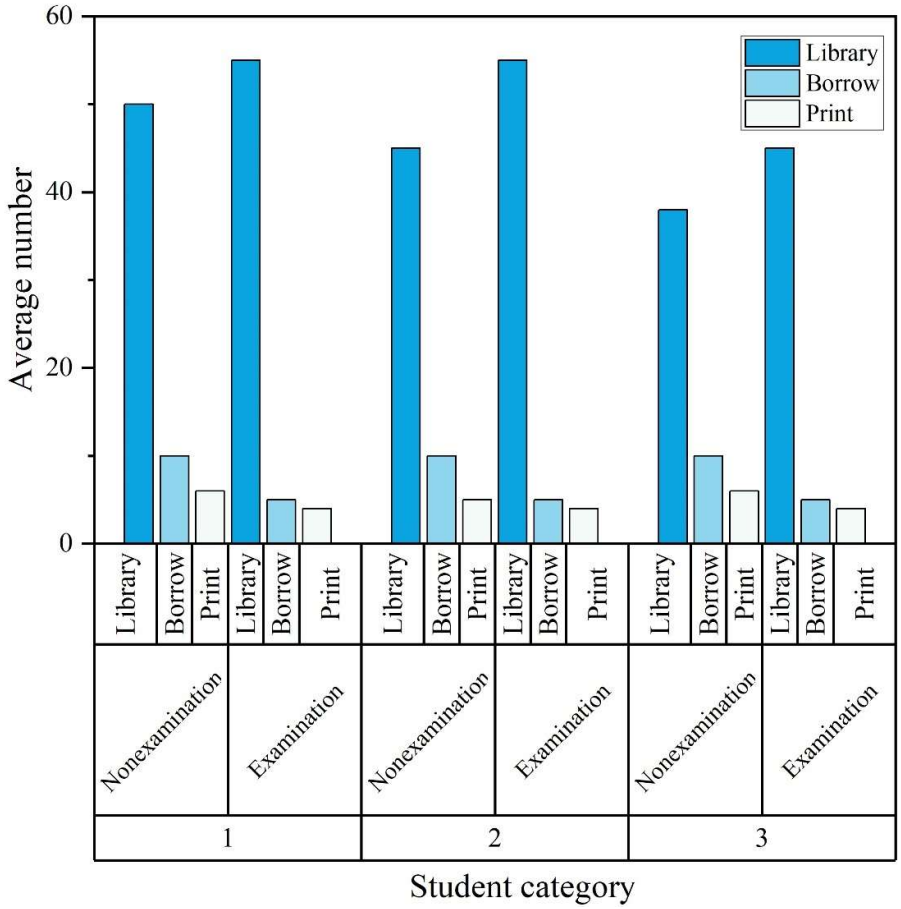

Fig. 7. Three kinds of students' behavior in the exam and non-exam period

3.3. Effectiveness of Student Behavior Prediction

3.3.1. Learning behavior prediction effects. In this paper, the student learning behavior data of a university from March 24 to March 25, 2023 is selected to construct the sample set, the consumption data from February 10 to March 23 in the sample set is used as the training set, and the established data set is trained based on the K-nearest-neighbor algorithm, and then the number of occurrences of the number of people who have behaved is predicted in the test set constructed by the data of the student learning behavior from March 24 to March 25. Here the experimental analysis is carried out by predicting the number of learning behaviors in the next time period through the first 42 time periods (one day), and the experimental effect graph of the predicted value of the number of learners after back-normalization is shown in Fig. 8, and from the effect graph of the prediction of the number of learners it can be seen that there is a better effect on the prediction of the number of learners on March 24th, 2023 through the model based on the K nearest neighbor algorithm, and the number of people predicted thereof is close to the real number of learners, and the number of learners predicted in the The whole time period in the predicted value and the real number of learners change trend is consistent, the real value and the predicted value of the number of learners have two peaks, respectively, 2023/3/24 08:00 and 2023/3/24 11:00 the predicted value and the real value of these two time periods were 2047, 1838, 1678, 1808, respectively.

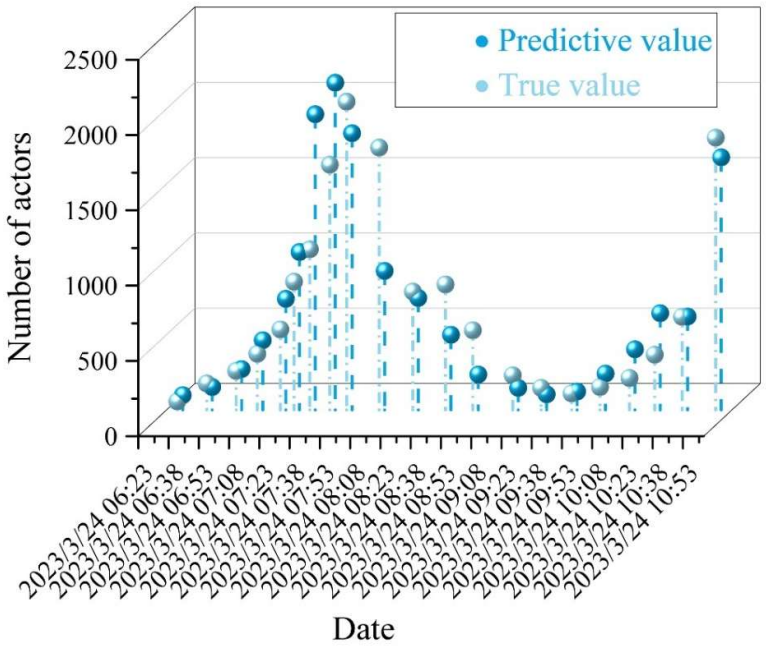

Fig. 8. Learning behavior prediction effect

Figure 9 shows the relative error of the prediction of the number of learners, the relative error of the number of learners less than 0.2 accounted for 86.42% of the sequence values, and the relative error of the number of learners less than 0.1 accounted for 71.6% of the sequence values, which means that more than 70% of the prediction of the number of learners has a greater than 85% accuracy. From the relative error chart of the prediction of the number of learners, it can be seen that the relative error of only five predicted values is greater than 0.4, which is mainly due to the fact that the model only takes into account the temporal characteristics of the number of learners and does not consider other factors. Although the relative error of the predicted number of learners in a few moments is large, the overall prediction of the number of learners is good.

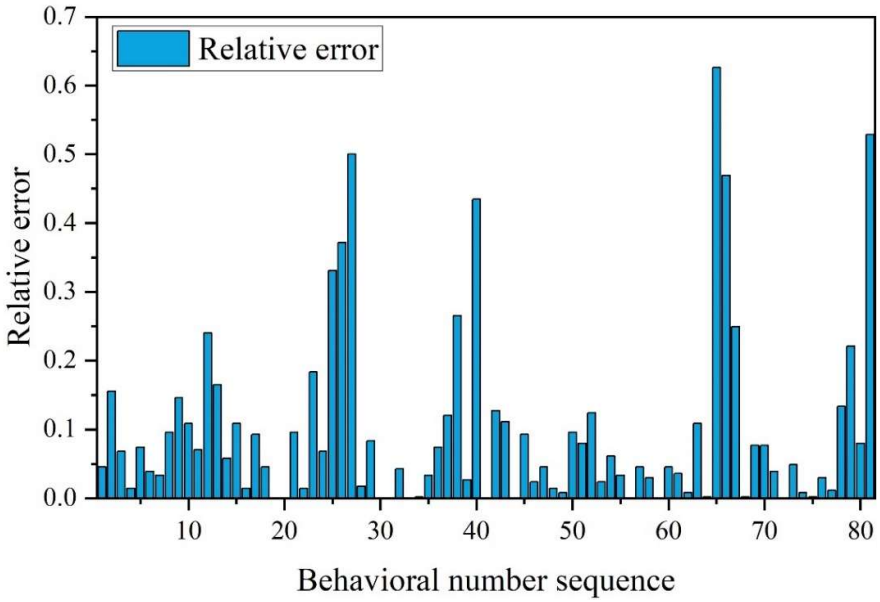

Fig. 9. The number of students predicted relative error

3.3.2. Comparison of Spark Cluster Based Acceleration. The parallelized implementation of the parallel H-mine algorithm on Spark is analyzed in terms of time efficiency. This paper evaluates the parallelism of the early warning model in this paper by comparatively analyzing the speedup ratio of the algorithm on different numbers of cluster worker nodes for different data volumes. The 10%, 50%, and 100% student data are used as the dataset, respectively, and the number of worker nodes in the Spark cluster is gradually increased to train the early warning model of this paper, and the results of the change in the acceleration ratio with the increase in the number of worker nodes are shown in Fig. 10.

With the increase of worker nodes, the acceleration ratio of the algorithm is improved. However, the actual parallelization of the algorithm because of the division of Stage, the existence of data width dependence between tasks leads to the impossibility of achieving 100 percent parallelism, and at the same time, because of the data dependence and communication between cluster nodes, the running time increases with the increase in the amount of data. When the amount of data is small (10%), the task submission and resource allocation of the Spark cluster and the communication between nodes take up too much time to reflect the benefits of parallelization. When the data volume is 100%, the acceleration ratio increases with the number of nodes, and reaches 4.596s when the number of worker nodes is 6. This also shows that Spark is more suitable for cluster computation of cluster computing volume.

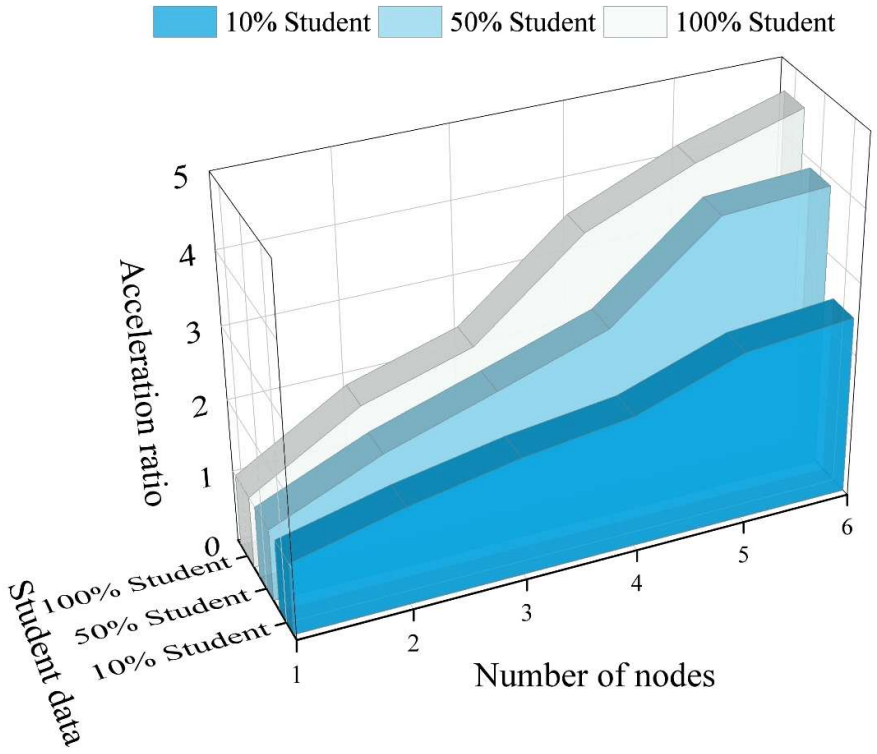

Fig. 10. Speed up on Spark cluster

4. Cluster computing-based student behavior analysis and management strategies

4.1. Optimizing the allocation of school resources

In-depth analysis of attendance data of different grades, majors, and classes reveals the reasons behind low attendance in certain courses, such as lack of student interest or irrational curriculum, and then adjusts the course schedule and teacher resources based on this, improves teaching

methods, and optimizes course content. Schools can also utilize cluster computing to identify students who are chronically absent or whose attendance is unstable, and by establishing an early warning system, generate alert messages in real time to remind teachers and administrators to intervene. Such automated monitoring can effectively reduce the burden on teachers and improve their efficiency in identifying and dealing with problems.

The application of cluster computing is also reflected in the dynamic allocation of resources. Based on attendance data, schools can flexibly adjust teacher staffing and classroom resources. Finally, schools can build attendance prediction models based on the analysis of multiple factors to foresee attendance trends in future semesters, providing a scientific basis for resource allocation and educational intervention strategies. If a course with low attendance is predicted, the school can take measures to enhance student interest or adjust the course design.

4.2. Monitor learning progress and provide instructional feedback

Teachers can use cluster computing analytics to uncover students' individual learning needs. Each student learns at a different pace and in a different way, and some students may be slower to master specific knowledge points, while others may have exceeded course requirements. Teachers need to carefully analyze student learning data to customize learning plans for students with different needs, such as providing additional resources or one-on-one tutoring for slower progressing students, while setting more challenging tasks for faster progressing students. Real-time monitoring of learning progress not only helps teachers to teach, but also helps students to improve themselves. In this regard, schools can set up a self-feedback system for students to view their own learning data at any time, so that they can have a clear understanding of their personal learning status and adjust their learning strategies in a timely manner to improve their learning results.

4.3. Personalized Learning Paths

Schools can apply the LMS to understand each student's performance in a particular course unit to automatically adjust the level of difficulty of subsequent courses. For example, students with high performance are provided with more advanced course content. In addition, schools should apply the LMS to understand students' interests and career development goals, and recommend appropriate elective courses and learning materials for them, so that they can explore knowledge in greater depth driven by their interests. To this end, schools need to provide diversified learning resources, such as e-textbooks, multimedia tutorials, online exercise libraries and virtual laboratories. Schools need to conduct accurate data analysis to identify and push resources that best suit the needs of each student. At the same time, schools can make use of cluster computing analytics to identify students who excel in specific subjects and provide them with more learning opportunities, such as recommending them to participate in high-level competitions or providing them with advanced curricular resources to further enhance their development.

4.4. Data feedback

To improve the quality of extracurricular activities, schools can use cluster computing to analyze the feedback data of activities and optimize their organization and management. Feedback from activities submitted by students through campus apps can identify strengths and weaknesses of the activities, which can then be used to adjust the frequency of the activities, the scale of participation, or redesign the content of the activities. With comprehensive extracurricular activity data analysis, schools should predict students' future activity interests and participation trends and plan ahead to ensure that the activities are in line with students' interests and needs. In addition, schools should analyze students' social networks and interaction patterns to identify socially marginalized students and provide them with necessary social support. By analyzing the activity preferences of different student groups, schools can customize the content of extracurricular activities to ensure that the

needs of each group are met, thereby increasing student participation and satisfaction and promoting their holistic development.

4.5. Developing a Precision Mental Health Intervention Program

Schools can use cluster computing analytics to develop more precise mental health intervention plans. For each student identified as high-risk, schools can develop interventions based on their individual data, such as arranging regular counseling, facilitating communication with parents and teachers, and recommending participation in relaxation and stress reduction activities. At the same time, schools can also analyze students' responses to various mental health support measures through cluster computing to determine which measures are most effective for the student population and optimize intervention strategies accordingly. To improve the effectiveness of interventions, schools can also use data to monitor the behavioral changes of students after receiving interventions. For example, data on students' attendance, grades, activity participation, and social interactions are tracked on an ongoing basis to assess the effectiveness of interventions in real time. If the data show improvement in the student's mental health, the current intervention program can be maintained. If the intervention is not effective, the strategy should be adjusted in a timely manner. In addition, schools should evaluate and improve mental health support services as a whole. Schools should analyze the data of the whole school to identify common problems and trends affecting students' mental health, and adjust the curriculum arrangement accordingly to strengthen mental health education.

5. Conclusion

This paper proposes a parallel H-mine cluster computing model based on Spark, mining the behavioral feature data of students in frequent items, combining information entropy and density optimization, performing clustering operation on the mined behavioral feature data, constructing a student behavior prediction model, and realizing the prediction and early warning of students' behaviors based on Spark clusters.

1) Using K-Means clustering algorithm, create portraits to label their attributes, and perform cluster analysis for student behavior, from the comprehensive score, the subject part of the students' scores are between 64 and 90 points, which is highly representative.

2) Set up simulation experiments, under nine different degrees of support, mining frequent items corresponding to the number of behaviors, set the minimum degree of support to 0.1, at this time there are a total of 105 frequent items, indicating that the parallel H-mine algorithm can achieve the analysis of student behavior.

3) Construct the data set of students' learning behaviors and use the K-nearest neighbor algorithm for training to predict the number of students' learning behaviors. The relative error of the predicted number of students is less than 0.2 in 86.42% of the sequence values of the number of students, and the relative error is less than 0.1 in 71.6% of the sequence values of the number of students, indicating that more than 70% of the predicted values of the number of students have more than 85% accuracy, and the model has good effect on the overall prediction of the number of students.

After analyzing the prediction of student behavior based on cluster computing, corresponding student behavior management strategies are proposed to adjust the behavioral strategies in time to improve the learning effect of students.

References

[1] Priyambada, S. A., & Usagawa, T. (2023). Two-layer ensemble prediction of students' performance using learning behavior and domain knowledge. Computers and Education: Artificial Intelligence, 5, 100149.

[2] Fu, Q. (2024). Research on Student Behavior Analysis and Grade Prediction System Based on Student Behavior Characteristics. Scalable Computing: Practice and Experience, 25(1), 217-228.

[3] Qu, S., Li, K., Wu, B., Zhang, S., & Wang, Y. (2019). Predicting student achievement based on temporal learning behavior in MOOCs. Applied Sciences, 9(24), 5539.

[4] Yao, H., Lian, D., Cao, Y., Wu, Y., & Zhou, T. (2019). Predicting academic performance for college students: A campus behavior perspective. ACM Transactions on Intelligent Systems and Technology (TIST), 10(3), 1-21.

[5] Shayan, P., & van Zaanen, M. (2019). Predicting student performance from their behavior in learning management systems. International Journal of Information and Education Technology, 9(5), 337-341.

[6] Wang, X., Yu, X., Guo, L., Liu, F., & Xu, L. (2020). Student performance prediction with short-term sequential campus behaviors. Information, 11(4), 201.

[7] Lai, Y., Saab, N., & Admiraal, W. (2022). University students' use of mobile technology in self-directed language learning: Using the integrative model of behavior prediction. Computers & Education, 179, 104413.

[8] de la Fuente-Mella, H., Guzmán Gutiérrez, C., Crawford, K., Foschino, G., Crawford, B., Soto, R., ... & Elórtegui-Gómez, C. (2020). Analysis and prediction of engineering student behavior and their relation to academic performance using data analytics techniques. Applied Sciences, 10(20), 7114.

[9] Tang, S., Peterson, J., & Pardos, Z. (2017). Predictive modelling of student behaviour using granular large-scale action data. The Handbook of Learning Analytics, 223-233.

[10] Baig, A. R., & Jabeen, H. (2016). Big data analytics for behavior monitoring of students. Procedia Computer Science, 82, 43-48.

[11] Feng, G., & Fan, M. (2024). Research on learning behavior patterns from the perspective of educational data mining: Evaluation, prediction and visualization. Expert Systems with Applications, 237, 121555.

[12] Wang, Y., Wen, J., Zhou, W., Wu, Q., Wei, Y., Li, H., & Tao, B. (2022). Research on Abnormal Behavior Prediction by Integrating Multiple Indexes of Student Behavior and Text Information in Big Data Environment. Wireless Communications and Mobile Computing, 2022(1), 1902155.

[13] Liu, X., Shao, M., Zhang, S., & Li, G. (2020, December). Research on Application of University Student Behavior Model Based on Big Data Technology. In 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE) (pp. 1387-1390). IEEE.

[14] Tu, L. (2019). Analysis and prediction method of student behavior mining based on campus big data. In Advanced Hybrid Information Processing: Third EAI International Conference, ADHIP 2019, Nanjing, China, September 21–22, 2019, Proceedings, Part II (pp. 363-371). Springer International Publishing.

[15] Han, L. (2024, May). Prediction and analysis of students' behavior based on data mining in educational administration. In International Conference on Artificial Intelligence for Society (pp. 229-238). Cham: Springer Nature Switzerland.

[16] Xian, X., Chen, F., & Wang, J. (2017). An insight into campus network user behaviour analysis decision system. International Journal of Embedded Systems, 9(1), 3-11.

[17] Liu, H., & Jiao, N. (2020). Research on Students' Campus Behavior Analysis and Warning System Based on Big Data. In Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery: Volume 2 (pp. 392-400). Springer International Publishing.

[18] Lwande, C., Oboko, R., & Muchemi, L. (2021). Learner behavior prediction in a learning management system. Education and Information Technologies, 26(3), 2743-2766.

[19] Chen, L., Wang, L., & Zhou, Y. (2022). Research on data mining combination model analysis and performance prediction based on students' behavior characteristics. Mathematical Problems in Engineering, 2022(1), 7403037.

[20] Uddin, M. A., Alam, A., Tu, N. A., Islam, M. S., & Lee, Y. K. (2019). SIAT: A distributed video analytics framework for intelligent video surveillance. Symmetry, 11(7), 911.

[21] Li, Y., Qi, X., Saudagar, A. K. J., Badshah, A. M., Muhammad, K., & Liu, S. (2023). Student behavior recognition for interaction detection in the classroom environment. Image and Vision Computing, 136, 104726.

[22] Mo, J., Zhu, R., Yuan, H., Shou, Z., & Chen, L. (2023). Student behavior recognition based on multitask learning. Multimedia tools and applications, 82(12), 19091-19108.

[23] Ngoc Anh, B., Tung Son, N., Truong Lam, P., Phuong Chi, L., Huu Tuan, N., Cong Dat, N., ... & Van Dinh, T. (2019). A computer-vision based application for student behavior monitoring in classroom. Applied Sciences, 9(22), 4729.

[24] Zhao, X. M., Yusop, F. D. B., Liu, H. C., Prilanita, Y. N., & Chang, Y. X. (2025). Classroom Student Behavior Recognition Using An Intelligent Sensing Framework. IEEE Access.

[25] Yin Albert, C. C., Sun, Y., Li, G., Peng, J., Ran, F., Wang, Z., & Zhou, J. (2022). Identifying and monitoring students' classroom learning behavior based on multisource information. Mobile Information Systems, 2022(1), 9903342.

[26] Chonggao, P. (2021). Simulation of student classroom behavior recognition based on cluster analysis and random forest algorithm. Journal of Intelligent & Fuzzy Systems, 40(2), 2421-2431.

[27] Wu, B., Wang, C., Huang, W., Huang, D., & Peng, H. (2021). Recognition of Student Classroom Behaviors Based on Moving Target Detection. Traitement du Signal, 38(1).

[28] Xu, T., Deng, W., Zhang, S., Wei, Y., & Liu, Q. (2023). Research on recognition and analysis of teacher–student behavior based on a blended synchronous classroom. Applied Sciences, 13(6), 3432.

[29] Liu, Q., Jiang, X., & Jiang, R. (2025). Classroom Behavior Recognition Using Computer Vision: A Systematic Review. Sensors, 25(2), 373.

[30] Arthanari, J., & Baskaran, R. (2019). Enhancement of video streaming analysis using cluster-computing framework. Cluster Computing, 22(Suppl 2), 3771-3781.

[31] Czyżewski, A., Bratoszewski, P., Ciarkowski, A., Cichowski, J., Lisowski, K., Szczodrak, M., ... & Krawczyk, H. (2015). Massive surveillance data processing with supercomputing cluster. Information Sciences, 296, 322-344.

[32] Patrikar, D. R., & Parate, M. R. (2022). Anomaly detection using edge computing in video surveillance system. International Journal of Multimedia Information Retrieval, 11(2), 85-110.

[33] Tsakanikas, V., & Dagiuklas, T. (2018). Video surveillance systems-current status and future trends. Computers & Electrical Engineering, 70, 736-753.

[34] Eickholt, J., & Shrestha, S. (2017, March). Teaching big data and cloud computing with a physical cluster. In Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (pp. 177-181).

[35] Jain, S., Ananthanarayanan, G., Jiang, J., Shu, Y., & Gonzalez, J. (2019, February). Scaling video analytics systems to large camera deployments. In Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications (pp. 9-14).

[36] Xu Yan,Wang Mingyu & Fan Wen. (2021). Defect Data Association Analysis of the Secondary System

Based on AFWA-H-Mine. Energies,14(14),4228-4228.

[37] Yuxiao Deng,Jingyu Liu & Yang Zhou. (2024). Research on Image Processing Resource Reconstruction Based on Load Balancing Strategy. Electronics,13(6),

[38] Kai Song,Haiming Zhang,Huaqiong Ma,Yongjun Sun & Liping Yan. (2025). Design and implementation of parallel k-means algorithm based on ternary optical computer.The Journal of Supercomputing,81(4),536-536.

[39] Tiantian Zhao,Long Yu,Xinyue Jiang,Shaoxiong Yang,Xuehui Zhang,Min Zhang... & Fang Xu. (2025). Construction of a public health practice teaching quality evaluation system based on the CIPP model using the Analytic Hierarchy Process (AHP) and entropy weight method. BMC Medical Education,25(1),368-368.