https://doi.org/10.31449/inf.v48i20.6779 Informatica **48** (2024) 29–40 **29**

Distributed Intelligent Optimization of E-commerce User Purchase Data Mining using Spark Framework

Jianjun Wu
College of Digital Economics and Trade, Kaifeng University Kaifeng 475004, China
E-mail: hnkfu8@163.com

Keywords: data mining, distributed algorithms, user purchase behavior, fitness values, spark computing framework

Received: July 25, 2024

As the development of e-commerce becomes more and more intelligent, higher requirements have been put forward for the algorithms controlling e-commerce operations. However, the current e-commerce operation is not timely and accurate enough to update the purchase data and statistics, resulting in cost consumption and revenue is not proportional, and can not accurately meet the user favorite. To speed up the collection of user purchase behavior data and improve the revenue of e-commerce operations, the study introduces adaptive degree values based on a distributed computing framework combined with a topological structure. The computing framework is used to speed up the calculation and convergence of user data, and the topology is responsible for classifying the data in the dataset and calculating the optimal location. The improved algorithm under the control of the topology structure, the accuracy of the product is above 94%, the highest is above 98%, compared with other algorithms, the accuracy is higher. The data collected on JD shopping platform shows that compared with other algorithms, the improved algorithm is improved by 81.2% due to the stability of the fitness value. In the simulation experiment, the overlap between the noise value of beauty search 2000-2700 and the noise value of clothing matching 2000-2500 in the shopping platform was large. Therefore, there was a correlation between the user's search for clothing collocation and the beauty search. In summary the improved algorithm is highly effective in both stability, accuracy, and applied error control. Therefore, the study of the improved algorithm has a better application for data mining of user purchase behavior.

Povzetek: Članek opisuje izboljšan algoritem za rudarjenje podatkov o nakupih uporabnikov v e-trgovini z uporabo Spark ogrodja.

1 Introduction

Today, with the rapid development of science and technology, intelligence has penetrated into all aspects of life. Especially, there are higher requirements for the intelligence of data mining and feature recognition [1-2]. Intelligence is even more central in the e-commerce industry. Stores use intelligence to create product links, optimize store pages, and so on. For the platform, intelligence controls the big data push of e-commerce. Such a way can be timely for the user preferences of the update [3-4]. For users, searching for products of interest in the search engine of the shopping platform also provides a channel for the cloud service platform to collect user preferences. The service platform stores data based on the total information provided by the search engine, with the objective of facilitating the next search conducted by the user by directly outputting similar data information [5]. However, if the data collection is not timely or comprehensive enough, it will result in the user's favorite information and purchase direction not being updated in time, which will lead to the impact on the store's operation. The current intelligent optimization algorithm has the problems of slow data mining rate and insufficient timely and comprehensive data collection. In this context, in order to improve the accuracy as well as the rate of data collection, the study designs optimization algorithms based on distributed computing framework (spark) combined with topology, which makes the data more regular through topology and introduces fitness values to adjust the data parameters. The spark framework can speed up the computation of the optimal position for data classification. It is expected that the optimized algorithm of the research can play a role in the data mining process of user's purchasing behavior.

The study is divided into four parts to analyze and design the algorithms. The first part is the analysis and comparison of domestic and international algorithms on data mining. The second part is the optimization of the core structure of the algorithm as well as the computing framework for the process research of data mining and user favoritism prediction. The third part is the performance test and application effect analysis of the improved algorithm. The fourth part is the summary discussion on the performance and application of the improved algorithm as well as its shortcomings.

2 Related works

Researchers at home and abroad have done a lot of research on mining data using algorithms. Yue et al. addressed the problem of data privacy leakage in traditional distributed algorithms. A distributed algorithm based on privacy protection was proposed, and the method

could also be applied to the problem of privacy optimization of industrial processes between different enterprises. The results proved that the proposed method improved data security [6]. Zhang Z et al. proposed a distributed decision intelligence framework based on evolutionary game theory for the task allocation problem of unmanned aircraft swarm systems in uncertain scenarios. Experiments proved that the proposed framework could effectively solve the task allocation problem [7]. Zhang et al. proposed a consensus-based distributed projection algorithm for the unbalanced problem of auxiliary vector computation graphs, which relied on local computation. The results proved the convergence of the algorithm on consistent joint strongly connected unbalanced directed graphs with different local constraints [8]. In response to the limitations of existing control methods for hybrid stacked girder cable-stayed bridges, Da et al. proposed a distributed algorithm and control method for cable force. This approach was developed to address the shortcomings of the existing permanent load balance method, influence matrix method, and adaptive control method for controlling electrons. The method adopted the constant load balance method and was adjusted according to the influence matrix of the main girder displacement. The method was found to be computationally efficient and to control the cable force with high efficiency and accuracy [9]. Shuang et al. designed an online optimization algorithm based on distributed mirror descent and distributed average tracking techniques for the distributed online optimization problem in an unknown dynamic environment. The method analyzed the dynamic regret and obtained the bounded regret. Simulation results verified the effectiveness of the designed algorithm [10].

Boujelben et al. proposed a fully distributed implementation technique for the quality of service and experience problems associated with distributed algorithmic control of cloud games. The family used an auction algorithm and several efficient extensions to solve the virtual machine placement problem. The experimental results proved that the proposed algorithm obtained significant improvement in terms of quality of service [11]. Anitha and Sumathi proposed a novel consensus-based anomaly detection method for wireless network sensors of IoT model, which are vulnerable to attacks. The data sensitivity was improved by the consensus feature and the experiments proved that the proposed method was more capable for attack defense [12]. In non-cooperative games, the problem that each participant's local objective function depends on both their own decisions and the decisions of other participants. To address this problem, Zheng et al. came out with a fixed step-size distributed generalized Nash equilibrium seeking algorithm based on backward reflection-forward backward splitting. Each player performed a backward step and then a forward backward step, and the pseudo-gradient was evaluated on the reflection term. Simulation results verified the effectiveness of the algorithm and the correctness of the theoretical analysis [13]. The limited computational resources of edge computing servers and the mobility of vehicles make the design of offloading strategies a very challenging problem. To address this problem, Shuang et al. proposed an efficient offloading and resource allocation scheme for network computing. The scheme adopted two offloading modes. Simulation experiments demonstrated the effectiveness of the proposed scheme [14]. For the energy management in multi-microgrid systems with pairwise problem and variable substitution, Lou and Fujimura proposed a fully distributed scheduling algorithm based on alternating direction multiplier (ADMM) and average-consensus (AC) as a way to solve the global optimal scheduling scheme. The results proved that the algorithm could effectively solve the problems in energy management [15]. Yan et al. proposed a stochastic gradient-free online distributed algorithm containing a multipoint gradient estimator for the case where the real gradient of the cost function generation of a multi-intelligent system was not available. Under this algorithm, the nodes made decisions using only the estimated information of the gradient instead of the real gradient information. Experimental results demonstrated the reliability of the algorithm [16]. An overview of the algorithms in the provided literature in the study is shown in Table 1.

Table 1: An overview of the algorithms in the provided literature in the study.

Network parameter	Method Name	Data Set Type	Key Metrics	Limitations of Existing Methods
Yue et al. [6]	Distributed Privacy Protection Algorithm	Industrial Process Data	Data Security Improvement	Not Mentioned
Zhang et al. [7]	Distributed Decision Intelligence Framework	UAV Swarm Task Allocation	Task Allocation Efficiency	Applicability in Uncertain Scenarios
Zhang et al. [8]	Consensus-based Distributed Projection Algorithm	Auxiliary Vector Computation Graphs	Algorithm Convergence	Dependence on Local Computation, Potential Computational Limitations
Da et al. [9]	Distributed Algorithm and Control Method	Hybrid Stacked Girder Cable-Stayed Bridges	Control Efficiency and Accuracy	Improvements on Existing Control Methods, Potential

				Limitations in Specific Scenarios
Shuang et al. [10]	Online Optimization Algorithm	Dynamic Environment	Dynamic Regret (Dynamic Regret)	Applicable to Unknown Dynamic Environments, Stability Not Mentioned
Boujelben et al. [11]	Distributed Implementation Technique	Cloud Gaming Quality of Service	Quality of Service Improvement	Virtual Machine Placement Problem, Potential for Scalability Limitations
Anitha [12]	Consensus-based Anomaly Detection Method	IoT Wireless Network Sensors	Data Sensitivity Improvement	Defense Against Attacks, Potential Network-Speci
Zheng et al. [13]	Distributed Generalized Nash Equilibrium Seeking Algorithm	Non-Cooperative Games	Algorithm Effectiveness and Theoretical Analysis Correctness	Dependence on Fixed Step Size, Potential Convergence Speed Limitations
Shuang et al. [14]	Resource Allocation Scheme	Network Computing	Efficiency	Targeting Edge Computing Servers and Vehicle Mobility, Potential Resource Allocation Limitations
Lou and Fujimura [15]	ADMM and AC-based Distributed Scheduling Algorithm	Multi-Microgrid Systems	Global Optimal Scheduling Scheme	Specific Energy Management Problem, Potential Computational Complexity Limitations
Yan et al. [16]	Stochastic Gradient-Free Online Distributed Algorithm	Multi-Intelligent Systems	Reliability	In distributed systems, information needs to be exchanged between nodes for gradient estimation, which can increase communication overhead

At home and abroad, for data mining and data feature recognition, most of them are based on the consensus property of distributed algorithms to improve the sensitivity of data data or adopt the computing framework of fully distributed algorithms to expand the search range of data. However, there is a lack of research on the arithmetic of distributed algorithm framework, as well as the movement path of data in space and the training of algorithm parameters. Therefore, the study of optimization of distributed algorithms based on the spark framework combined with the introduction of fitness values for topology is very meaningful for user purchase data mining.

3 Modeling distributed optimization algorithms based on mining user purchase data

The study is divided into two sections on the optimization of the intelligent algorithm's data mining approach to purchase behavior. The first section is about the e-commerce data mining of the model of the optimized algorithm and the demonstration of the prediction process of the purchase behavior. The second section is an optimization study of the core structure of the intelligent algorithm.

3.1 Improving adaptive function and topology for data mining by intelligent distributed algorithms

The booming development of e-commerce makes the algorithm put forward more stringent requirements for traditional data mining. Based on the distributed technology computing framework, the data is partially processed through parallel computing, but its computing efficiency is much lower than the spark computing analysis framework. Moreover, the algorithm converges too early and the data search is incomplete. Therefore, the study optimizes the intelligent distributed algorithm based on the spark computing mechanism and reconstructs the distributed optimization algorithm model. The construction process of the model in e-commerce data mining is shown in Figure 1.

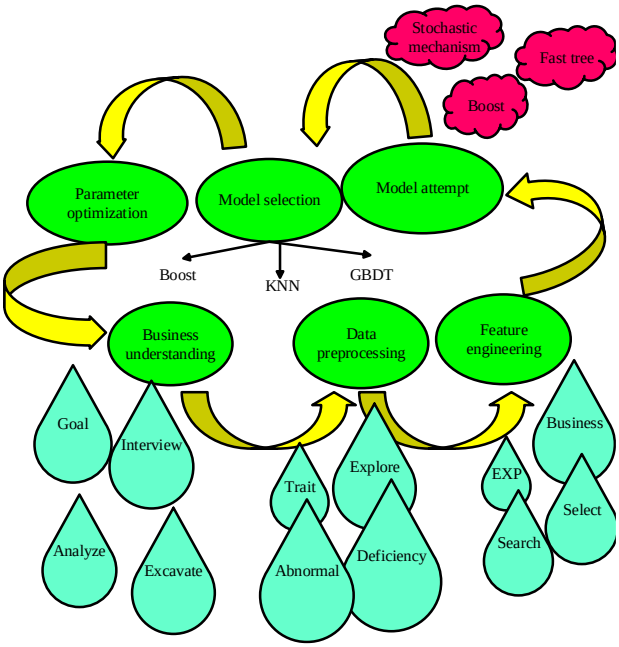


Figure 1: The construction process of model in e-commerce data mining.

In Figure 1, the optimization model needs to confirm the mining objective first in the process of e-commerce data mining. This step ensures that the decisiveness and accuracy of decision-making is free from errors and prevents losses and other unfavorable phenomena [17-18]. After the target is determined, the data initialization process is carried out. Data initialization process relies on the exploration of e-commerce data. The core steps are data exploration and missing value processing. In order to facilitate the multiple utilization of data, the topology is used to increase the information exchange between the data and prevent the incomplete search caused by the algorithm converging too fast. The last thing is to optimize the parameters of the model. The adaptive model parameters constructed by the study can autonomously perform parameter tuning to achieve the optimal solution of the model. The adaptive objective function can be expressed by Equation (1).

$$goal(x) = \min J_m(U,c) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_y^m d_{ij}^2 \tag{1}$$

In Equation (1), $J_m(U,c)$ denotes the functional representation of the spatial coordinates of the model. C and n denote the training coefficients. u_y^m denotes the spatial domain. d denotes the number of parameters debugging times. i and j denote the parameter coefficients. The constraints can be expressed in Equation (2).

$$\sum_{i=1}^{c} u_{ij} = 1, 1 \le j \le n \tag{2}$$

The spatial domain of the constraints can be expressed in Equation (3).

$$u_{ij} \in [0,1], 1 \le j \le n, 1 \le i \le c \tag{3}$$

The relationship between the spatial domain and the number of training sessions can be expressed in Equation (4).

$$0 \le \sum_{j=1}^{n} u_{ij} \prec n, 1 \le i \le c \tag{4}$$

In Eqs. (2), (3) and (4), n and C denote the number of prediction optimizations. j denotes the optimal number of training times adapted to function adjustment. i denotes the maximum number of training times when no function is introduced. The center vector of the overall data sample can be expressed in Equation (5).

$$x' = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m x_j}{n} \tag{5}$$

In Equation (5), m denotes the number of useful samples. n denotes the total number of samples. The adaptive function can be expressed in Equation (6).

$$L(c) = \frac{\sum_{i=1}^{c}\left(\sum_{j=1}^{n} u_{ij}^m\right)\|v_i - x\|^2/(c-1)}{\sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m \|x_j - v_i\|^2/(n-c)} \tag{6}$$

In Equation (6), v_i denotes the rate of data processing. X denotes the independent variable factor of the function. In the process of mining and processing e-commerce data information, the inclusion of the adaptive function combined with the memory-based spark computing and analysis framework speeds up the

algorithm's computing rate and the optimization process of the model parameters. In practice, the tuning of the parameters of the adaptive function and the discussion of the computational overhead of the topology have been supplemented. The specific content is as follows. The way of tuning the adaptive function parameters is to adjust the local difference of the initial distribution according to the number of links of the neurons, so as to ensure the same input and output variance of each neuron as much as possible. Then, the hyperparameters are automatically adjusted according to the performance feedback during the model training process. The influence of the algorithm topology on the computational overhead is mainly reflected in the network latency, power consumption, and the efficiency of data processing and transmission. Topology affects network latency by affecting the path length of message passing and the number of nodes passing through. Because information needs to go through routers and links during transmission, this process consumes energy. Therefore, the topology directly affects the power consumption of the network by affecting the path and jump number of data transmission. The topology also determines the total number of available paths between nodes, thereby affecting the network's ability to meet bandwidth requirements and process data. Different topological structures have different advantages and disadvantages in different application scenarios. The data trajectory of the improved algorithm incorporating the topological mechanism is shown in Figure 2.

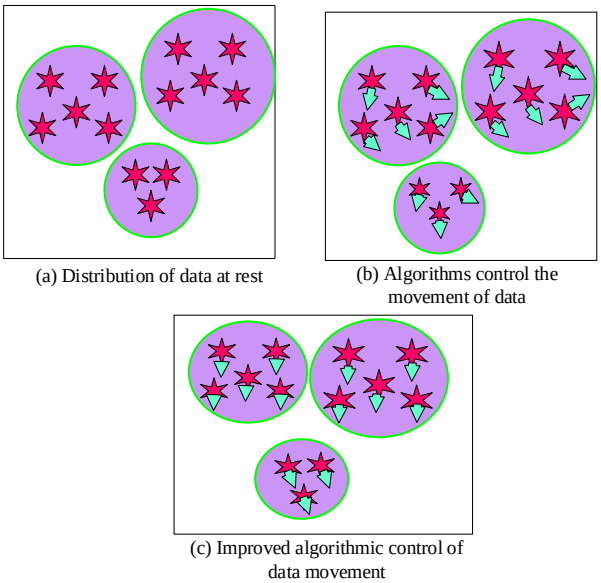


Figure 2: The data running trajectory of the improved algorithm integrated with the topological mechanism.

In Figure 2, the data information before it is incorporated into the topological mechanism is disorganized among themselves. The data moves in a directionless manner in space in radial directions that are not coherent. After incorporating the topology mechanism, the data starts moving in the same direction in an orderly manner. The data information in the topological structure belongs to a whole, so a change in the topological mechanism causes the data to move in a different direction as a whole. But the mechanism does not limit the tiny movements of the data solids in such a way that the overall randomness is ensured yet the local wholeness is ensured [19]. The velocity equation of the updated data set can be expressed as Equation (7).

$$V_{iD}^{t+1}=\omega*V_{iD}^{t}+C_1*L_1(\theta)*\left(d*Pbest_{iD}^{t}\right)+C_2*L_2(\theta)*\left(d*Gbest^{t}-X_{iD}^{t}+X_{iD}^{t}seed-X_{iD}^{t}\right) \tag{7}$$

In Equation (7), $X_{iD}^{t}seed$ expresses the center position of the group in which the data are located. t denotes the number of generations of data. i denotes the amount of data. D denotes the number of spatial dimensions. C denotes the structure factor. ω denotes the motion factor. $Gbest$ denotes global optimum. $Pbest$ denotes individual optimization. L denotes data set motion length. d denotes the interference factor. A nonlinear variation is used, the stronger the interference, the larger the search range will be, and the intensity of the interference will be reduced at a later stage. The interference factor can be expressed by Equation (8).

$$d=d_{\max}-\frac{d_{\max}-d_{\min}}{sum(1:T_{\max})}*(T_{\max}-t) \tag{8}$$

In Equation (8), $d_{\max}$ denotes the maximum value of the interference factor. $d_{\min}$ denotes the minimum value of the interference factor. $T_{\max}$ denotes the maximum value of motion space. Because the introduction of the topological mechanism leads to the linkage of topological relationships between individuals, the direction of movement and velocity of topological individuals between groups remains consistent. The individual velocity update after incorporating the topological mechanism can be expressed by Equation (9).

$$V_{iD}^{t+1}=\omega*V_{iD}^{t}+C_1*L_1(\theta)*\left(d*Pbest_{iD}^{t}\right)+C_2*L_2(\theta)*\left(d*Gbest^{t}-X_{iD}^{t}+X_{iD}^{t}seed-X_{iD}^{t}\right)+C_3*L_3*Vtopo\log y_{iD}^{t} \tag{9}$$

In Equation (9), C_3 denotes the topological factor. $Vtopo\log$ denotes the topological term. The speed after transformation can be optimized as Equation (10).

$$Vtopo\log y_{iD}^{t}=\frac{1}{N}\sum_{n\in T}V_n^{t} \tag{10}$$

In Equation (10), N denotes the total number. In general distribution algorithms, the movement of data is irregular and random. After adding the topology the movement of data becomes directional and regular, which

greatly improves the efficiency of the algorithm search. The flowchart of the optimized algorithm is shown in Figure 3.

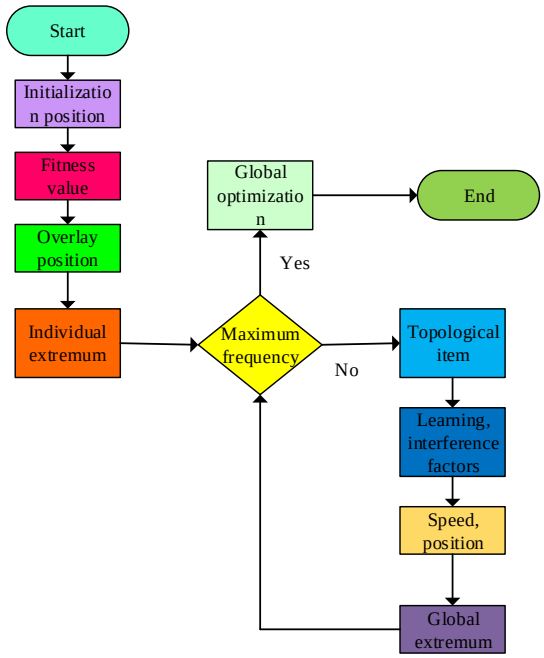


Figure 3: Optimized algorithm flow chart.

In the flowchart of Figure 3, the first step of this algorithm is to initialize the position and speed of the dataset. The second step is to evaluate the corresponding fitness values based on the current position or parameter values, and calculate the current fitness values of the data group and the fitness values of the reverse data group, respectively. The position of the reverse dataset was calculated using the equation (11). The third step is to calculate the overlapping position, and if there is an overlapping position, perform the following to find the best solution for the current position by individual extrem values. The fourth step is to optimize the global, determine whether the iterations stop condition; if yes, end the whole process, not update the speed of the data set using formula (9), and calculate the learning factor and weight interference factor of the data set. The fifth step is to compute the optimal topology in the current state. The sixth step is to update the speed and position according to the learning and interference factors, and then to update the speed and position of the particles to find a better solution. Finally, the best solution in all positions is found, and when the maximum frequency or other stopping conditions is reached, the algorithm ends and returns the final global optimal solution. From this function we derived the position of the dataset as well as the optimal value of fitness and the change position of the dataset after a Gaussian mutation. The update of movement speed, learning factor and fitness values are optimized to obtain the position of the reverse dataset, resulting in the final optimal position of the dataset.The update of the reverse data set can be expressed in Equation (11).

$$x'_k = x_{\min} + x_{\max} - x_k \tag{11}$$

In Equation (11), $x_{\min}$ and $x_{\max}$ denote the interval maximum and minimum values of individual positions of the data set. The adaptation value is denoted as $f(x)$. The reverse adaptation value is denoted as $f(x)'$. The two are compared and if $f(x)' \prec f(x)$, then $x' = x$. The weights of adaptive adjustment can be expressed in Equation (12).

$$\omega_i = k^{-\frac{fitness}{Gbest}} \tag{12}$$

In Equation (12), k denotes the weight coefficient. $fitness$ the adaptive minimum. $Gbest$ denotes the optimal position. The movement of the data set is made more regular through the topology, and the optimal position is derived through the calculation of the inverse function.

3.2 Spark computing framework and operating principles for distributed optimization algorithms

The study optimizes the intelligent distribution algorithm by introducing adaptive function and topology structure, which reduces the time consuming algorithm parameter optimization and improves the rigor of parameter optimization. However, the rate of the algorithm in the overall data mining and processing has not been significantly improved. Therefore, the research added the spark computing framework from the computational point of view. The structure of data in the framework is shown in Figure 4.

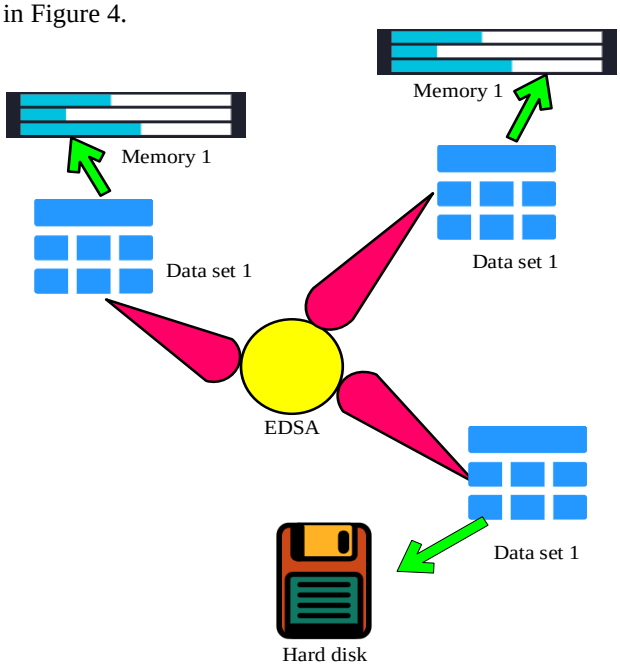


Figure 4: Data structure of user purchases in the framework.

In the data structure schematic shown in Figure 4, there are three ways to deploy the operational framework:

standalone mode, application mode, and resource management mode. The standalone mode has a complete service model that allows the data sets to be laid out independently without relying on other management systems for assistance. This mode is the core of the other modes of operation. The remaining two models are realized by system management for data computation and resource handling. In the system of the framework, the core concept is elastic data set distribution (EDSA) [20]. Reconstruction is done after data inappropriateness to ensure that the data carries the ability to repair itself. After mining based on the information of memory degree data, the support of content type can be represented by Equation (13).

$$T_{A_p \to B_p} = \frac{\left| M\left(A_p \cap B_p\right) \right|}{|M|} \tag{13}$$

In Equation (13), $A_p \cap B_p$ denotes the set of types of the dataset. M denotes the total amount of type data, and the dataset with high support items can be expressed as Equation (14).

$$\frac{|Y(X)|}{|M|} \geq T_{\min} \tag{14}$$

In Equation (14), $T_{\min}$ denotes the minimum value of support. X denotes item. Y denotes the item set. The confidence level of the dataset is found for the frequently viewed data is expressed in Equation (15).

$$\xi f = \frac{T_F}{T_S{}'} \tag{15}$$

In Equation (15), T_F denotes the confidence interval length. $T_S{}'$ denotes the length of the confidence interval for the non-empty subset. The predictive architecture for analyzing users making repeat purchases through memory-based operations is shown in Figure 5.

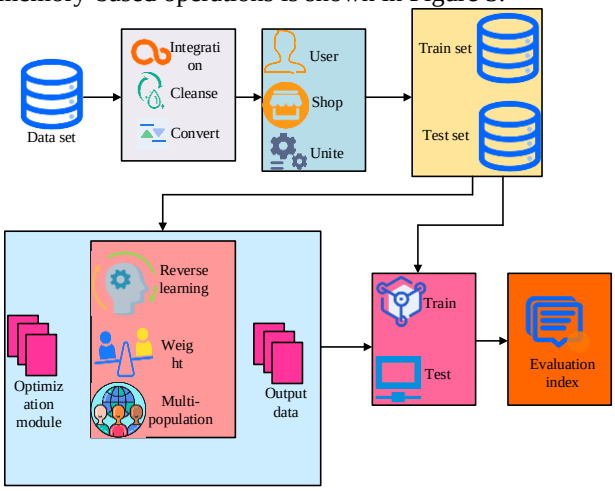


Figure 5: Predictive architecture in which users make repeated purchases.

On the prediction system of Figure 5, the raw dataset is first keyed from the input for data preprocessing. The preprocessing part includes data integration, data cleaning, and data transformation. The dataset features are processed through feature engineering is completed to start the dataset slicing. Purchase data through the training set and test set of the algorithm to realize the dataset slicing. In which the dataset needs to be tested by the improved strategy of distribution algorithm with hybrid model of spark computing framework. The data computed by the improved algorithm module is used to update the inertia weights as well as the fitness values through the function of the strategy of reverse learning and then output the dataset classification. The predicted data is finally passed through the evaluation function to produce an evaluation of the predicted behavior. The enhancement model of this evaluation metrics algorithmic framework is shown in Figure 6.

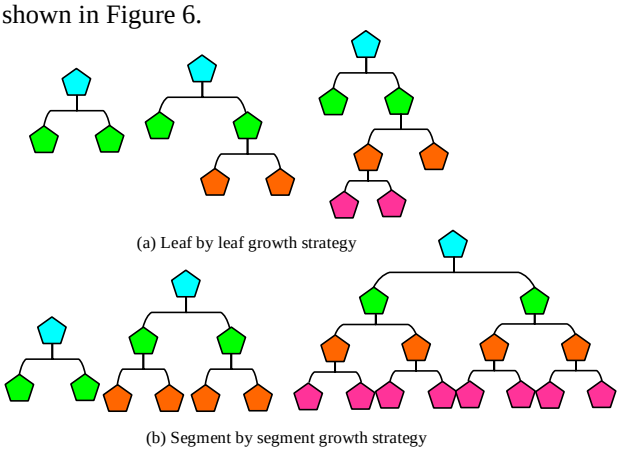


Figure 6: The upgrading model of evaluation index operation framework.

In Figure 6, the framework is optimized to effectively reduce the number of sample features. The optimal feature segmentation point is found by reducing the number of features. Figure 6(a) represents the leaf-by-leaf growth pattern of the frame. The samples grow leaf-by-leaf from the gradient-based side, and the samples are sorted according to the absolute value of the samples at the time of growth. The gains of smaller gradient samples are scaled up for computation, and larger gradient samples contribute more to the computation of gain values. Most of the strategies for the tree after this boosting take a Figure 6(b) level-by-level strategy. The leaf-by-leaf growth is then able to find the leaf with the largest gain from splitting the leaf from among all the leaves present and then re-split it. Such an operation reduces the occurrence of errors and improves the accuracy of the algorithm.

4 Performance test of the improved algorithm and analysis of the application effect

The study is divided into two sections to test the performance of the improved algorithm and analyze the effect of the application. The first section is the

performance test of the improved algorithm incorporating adaptive function combined with topology. The second section is about the analysis and study of the effect of the model incorporating adaptive function incorporating topology in practical applications.

4.1 Performance testing of improved algorithms

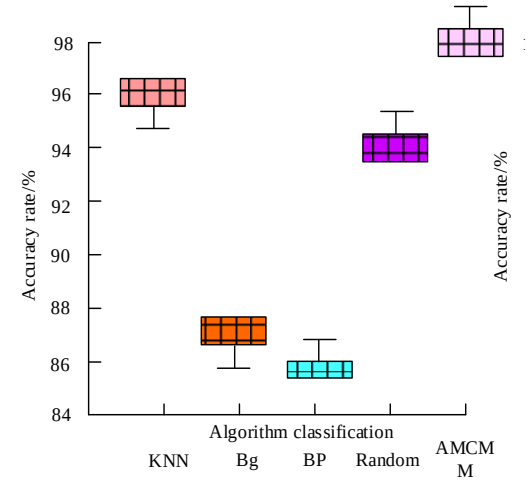
The e-commerce user purchase data as well as user store browsing information is processed by an improved algorithm that incorporates the fitness function and memory framework. The fitness function is used to plan the direction of motion of the data set and better solve the reverse data set. The combination of memory framework changes the traditional way of data computing and improves the speed of data processing. The experimental environment for performance testing of the improved algorithm is shown in Table 2.

Table 2: Experimental environment information.

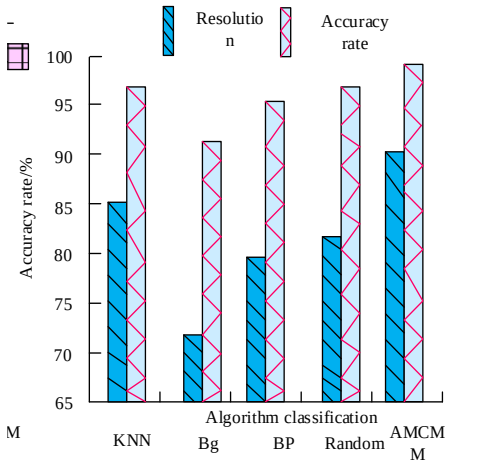
Designation	Version information	Environmental grade
Number of nodes	8	Primary
Internal memory	64G	Intermediate
CPU	Intel(R)Core(TM)i9-8980CN	Advanced
Linux	Centos9	Advanced
JDK	3.7.2	Advanced
Spark	1.6.0	Advanced
Hadoop	3.1.3	Advanced

Table 2 shows the specific requirements of the algorithm for the experimental environment. The preferred number of nodes for testing the algorithm is 8, and the software memory is 64G. The CPU is the more advanced Intel(R) Core (TM) i9-8980CN. Linux is the advanced Centos9. The JDK is the advanced version 3.7.2. The version of the in-memory algorithm is the version 1.6.0. Hadoop is version 3.1.3. Suitable version information as well as class requirements are essential for data mining as well as processing. The accuracy of different algorithms for categorizing user purchase data is shown in Figure 7.

In Figure 7(a), the accuracy of feature classification in the technology assessment box and line diagram of KNN algorithm is 96%. The accuracy of feature categorization in box-and-line diagram for technology assessment of Bg algorithm is 86%. The accuracy of feature classification in different classification algorithms of BP neural network is 86%. Accuracy of feature classification in stochastic algorithm is 95%.



(a) Evaluation boxplot of different algorithm techniques



(b) Comparison of recognition performance of different classifiers

Figure 7: The fitting curve of the original and predicted values of the structure thickness and the recognition performance of different classifiers.

Feature classification accuracy of improved algorithm of Adaptive function combined with memory mechanism (AMCMM) is 98%. Figure 7(b) shows the comparison of the recognition performance of different classifiers. The KNN algorithm achieves 85% feature resolution and 97% identity recognition accuracy. The Bg algorithm has a feature resolution of 73% and an identity recognition accuracy of 92%. The BP algorithm achieves 83% feature resolution and 96% identification accuracy. The stochastic algorithm has a feature resolution of 82.5% and an identity recognition accuracy of 95%. The feature resolution of AMCMM improved algorithm reaches 94% and the identification accuracy reaches 99%. The significance level of the improved algorithm is equal to 0.05, and the results are credible. The spatial variation of the data after mining by the algorithm is shown in Figure 8.

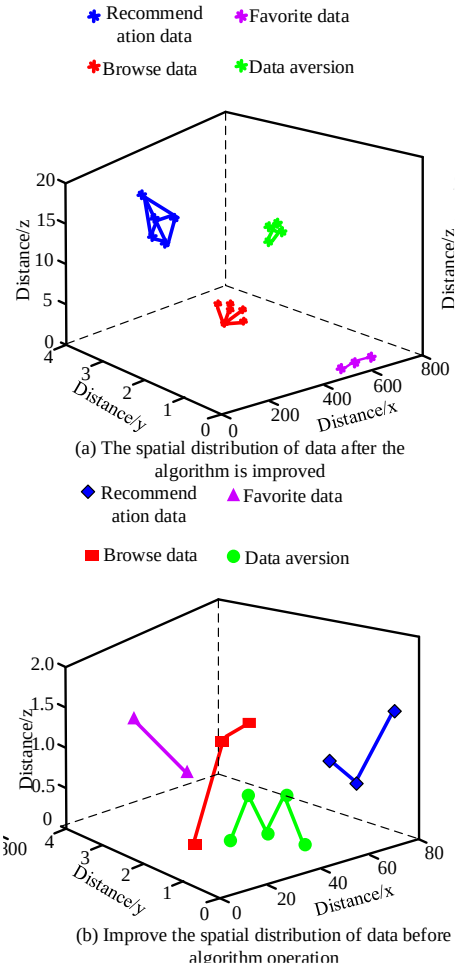


Figure 8: Spatial changes of data after algorithm mining.

In Figure 8, the data is categorized into platform recommendation data, user favorite data, user browsing data, and user disliked data based on the different attributes of the user data. Figure 8(b) shows the shape of distribution and existence of the data in space without being processed by the improved algorithm. The platform-recommended data is at a distance of 60 in the horizontal coordinate of space, with a vertical coordinate of 1 and a spatial coordinate of 0.5. The data groups are more dispersed from each other, with different movement trajectories. In Figure 8(a), the data processed by the improved algorithm form a compact data group in space. The distance between data individuals becomes smaller, the spatial horizontal coordinates become smaller, and both the vertical and spatial coordinates become larger. The other data distributions are similar to the case of the platform recommendation data. The distribution of data not processed by the improved algorithm is chaotic and irregular, and the distance between individual data is far away. The data processed by the improved algorithm becomes a tightly distributed data group, and the individual data are connected to each other.

4.2 Application effect of the improved algorithm

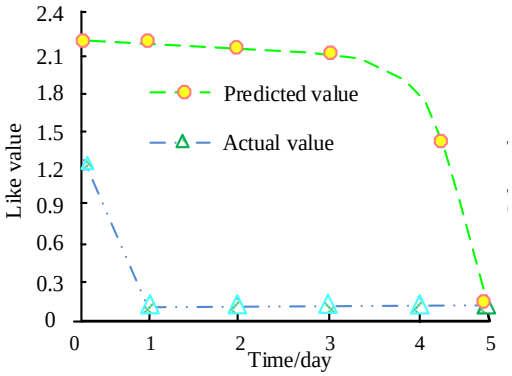
The improved algorithm has been tested and proved to have better accuracy in data processing and classification. Moreover, the data processed by the algorithm has better regularity and aggregation. The algorithm is applied to the shopping platform of Jingdong to collect information on users' purchasing behavior through the search engine. The parameters and initial values for analyzing user purchase data are shown in Table 3.

Table 3: Algorithm parameters and initial values.

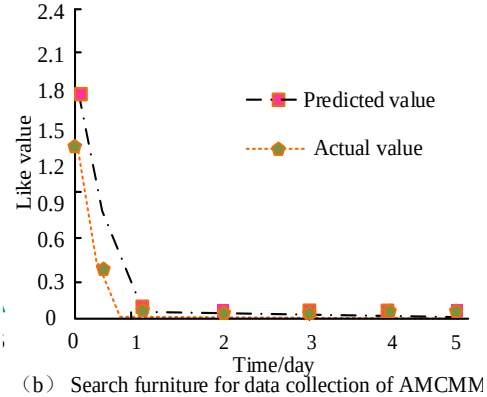
Parameter name	Version information	Parameter description
Leaf count	Maximum number of leaf nodes	150
Characteristic number	The maximum value of the feature number	9
Minimum number of samples	The minimum number of internal nodes is subdivided	Acquiesce
Minimum sample number of leaf nodes	Minimum number of leaf nodes	Acquiesce
Maximum depth	Depth of the decision tree	9
Value range	Sampling range	0.5
Step size	The weight coefficient of the learner	0.1
Number of iterations	The number of iterations of the weak learner	80
Data set	Iris data set: A categorical dataset for the testing of pattern recognition and machine learning algorithms	150 samples, each with four characteristics

In Table 3, the maximum number of iterations for the weak learner of the algorithm is set to 80 in the simulation experiments, which indicates that 80 iterations allow the parameters of the algorithm to be tuned to be optimal. At this time, the algorithm bias value is most reasonable. The maximum number of leaf nodes of the algorithm is 150, which meets the classification requirements for the mined data. The maximum number of features is 9. The minimum number of samples for internal nodes as well as the minimum number of samples for leaf nodes are default values. The maximum depth of the decision tree is 9, and the range of values without put-back sampling is 0.5. The weight reduction factor of each weak learner, that is, the step size of the algorithm is required to be 0.1. The data set information for the algorithm is Iris data set: A categorical dataset for the testing of pattern recognition and machine learning algorithms.The parameter information of the data set is150 samples, According to

the upgrade of the hardware of the improved algorithm and the dataset, the hardware environment was benchmarked, and it is found that in the hardware environment of Table 1, the confidence interval of 95% of the running time of the algorithm is in the high efficiency stage, indicating that 95% of the possible algorithm operation is kept in the high efficiency stage. Each with four characteristics. The algorithm is used in the Jingdong platform to collect users' search data for furniture after five days of application to analyze. The results are shown in Figure 9.



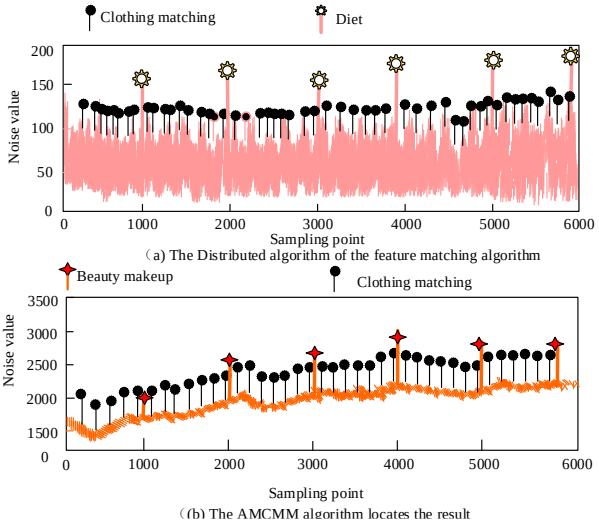
(a) Search furniture for data collection of DA



（b） Search furniture for data collection of AMCMM

Figure 9: Loss function curve of DA and AMCMM.

In Figure 9, the improved algorithm has a significantly better fitting relationship between the actual and predicted values for the user search furniture data, while the distributed algorithm has a larger deviation for the user search furniture data collection compared to the actual values. In Figure 9(a), the favorite value of the user searching for furniture on the first day is around 0.1, while the algorithm calculates the user's favorite value to be around 2.14. The AMCMM algorithm's calculation of the user's favorite value for searching furniture is almost the same as the actual value of 0.1. As the number of days of data collection by the algorithm increases, the user search index for furniture shows a decreasing trend. The relationship between user search for clothing and diet and cosmetics is shown in Figure 10.



（a) The Distributed algorithm of the feature matching algorithm

（(b) The AMCMM algorithm locates the result

Figure 10: Localization of original signal by similarity algorithm and matching algorithm.

In Figure 10, the distributed algorithm data collection generates a large amount of data traffic, but the collected data noise is small. As the test sample gets larger, the distributed algorithm continues to have data noise values between 100-150 for users searching for clothing matches. The improved algorithm sustains between 2000-2500. The noise values for users searching for clothing pairings who also searches for diet related at the same time ranged from 150-200. The noise values collected by the improved algorithm for user searches for beauty are between 2000-2700, indicating that there is a strong connection between users' purchasing behavior for clothing and beauty.

5 Discussion

The AMCMM algorithm can better maintain data integrity and richness when processing complex datasets, thus performing better in specific scenarios. Subdivision of accuracy improvement: feature binarization is a common data preprocessing technology, which is especially suitable for improving the processing efficiency of algorithms on sparse data sets. However, this approach may lose some information contained by continuous feature values, thus affecting classification accuracy. In contrast, the AMCMM algorithm optimizes the parameters by adaptive functions, maintains the original features of the data, and enhances the expression power of the data through the topology, thus achieving over 94% accuracy in classification accuracy, up to 98%. The impact of topology and Spark framework on computational efficiency: Topology is used in AMCMM algorithm to enhance the information exchange between data and prevent premature algorithm convergence, while the Spark framework accelerates the data processing speed through in-memory computing. The combination of these two significantly improves the computational efficiency of the algorithm, especially on large-scale datasets, where the AMCMM algorithm is able to achieve convergence faster compared to the additional processing steps that may be required after feature binarization. Computational

complexity and Performance Benefits: Although the AMCMM algorithm may introduce higher computational complexity in parameter optimization and topology, the performance gains are significant. Compared to the possible information loss caused by feature binarization, the AMCMM algorithm achieves high efficiency and high accuracy while maintaining data integrity, a trade-off that is reasonable in specific scenarios. Performance in a specific scenario: In a specific scenario of e-commerce user purchase behavior analysis, the diversity and complexity of the data requires the algorithm to be able to capture subtle pattern changes. The AMCMM algorithm is able to handle such complex datasets better through adaptive adjustment and topological structure optimization, while the feature binarization may lose key information by simplifying the data.

6 Conclusion

With the booming development of the e-commerce industry, reducing the operating costs of the store and increasing the revenue of the operation has become an urgent problem to be solved nowadays. In order to improve the enthusiasm of e-commerce operation, accelerate the construction of a new economic system, and reduce the time and labor cost of data processing, the study introduces the adaptive value based on the spark computing system and topology to improve the intelligent distribution algorithm. The method classifies the data regularly by topology and accelerates the rate of data mining based on the memory computing mechanism. In the algorithm data classification accuracy experiment, compared with the 73% and 88% data classification accuracy of other algorithms, the AMCMM algorithm's accuracy reached more than 94%, and the highest could reach 98%. In the data spatial distribution experiment, the data recommended by the platform was 60 in the distance of the spatial horizontal coordinate, and the vertical coordinate was 1, and the spatial coordinate was 0.5. It became more regular and directional after being processed by the algorithm. In the analysis of the application effect, the favorite value of the user searching for furniture was about 0.1, while the user's favorite value calculated by the algorithm was about 2.14. The AMCMM algorithm's calculation of the user's favorite value for searching furniture was almost the same as the actual value of 0.1. The noise value of cosmetics searches in shopping platforms was 2000-2700, which overlapped with the noise value of 2000-2500 with clothing matching, so there was a correlation between users' searches for clothing matching and cosmetics searches. In summary, the improved algorithm has good stability, high accuracy and low error, which has application value for data mining. However, the study has not analyzed the impact of other disturbing factors on the data mining process.

References

[1] Sandip Garai, Ranjit Kumar Paul, Mohit Kumar, Anish Choudhury, Intra-Annual National Statistical Accounts Based on Machine Learning Algorithm. Journal of Data Science and Intelligent Systems, 2(2): 12-15, 2023. https://doi.org/10.47852/bonviewJDSIS3202870

[2] Narender Chinthamu, Manideep Karukuri, Data Science and Applications. Journal of Data Science and Intelligent Systems, 1(1): 83-91, 2023. https://doi.org/10.47852/bonviewJDSIS3202837

[3] V. T. Ram Pavan Kumar, M. Arulselvi, K. B. S. Sastry. Comparative Assessment of Colon Cancer Classification Using Diverse Deep Learning Approaches. Journal of Data Science and Intelligent Systems, 1(2): 128-135, 2023. https://doi.org/10.47852/bonviewJDSIS32021193

[4] Jiayi Fan, Wenjing Xu, Yi Huan, R. Dinesh Jackson Samuel. Application of chaos cuckoo search algorithm in computer vision technology. Soft Computing, 25(18): 12373-12387, 2021. https://www1.cs.columbia.edu/CAVE/software/softlib/coil-100.php.

[5] Juan Li, Yuan-Hua Yang, Hong Lei, Gai-Ge Wang. Solving logistics distribution center location with improved cuckoo search algorithm. International Journal of Computational Intelligence Systems, 14(1): 676-692, 2021. https://doi.org/10.2991/ijcis.d.201216.002

[6] Changyang Yue, Wenli Du, Zhongmei Li, Bing Liu, Rong Nie, Feng Qian. Differential privacy distributed optimization algorithm against adversarial attacks for efficiency optimization of complex industrial processes. Advanced Engineering Informatics, 62 (PB): 102662-102665, 2024. https://doi.org/10.1016/j.aei.2024.102662

[7] Zhe Zhang, Ju Jiang, Haiyan Xu, Wen-An Zhang. Distributed dynamic task allocation for unmanned aerial vehicle swarm systems: A networked evolutionary game-theoretic approach. Chinese Journal of Aeronautics, 37(6): 182-204, 2024. https://doi.org/10.1016/j.cja.2023.12.027

[8] Yichen Zhang, Yutao Tang, Zhipeng Tu, Yiguang Hong. Distributed algorithm for solving variational inequalities over time-varying unbalanced digraphs. Control Theory and Technology, 22(3): 431-441, 2024. https://doi.org/10.1007/s11768-024-00223-9.

[9] Da Wang, Lei Wang, Zheng Li, Shengtao Xiang. Optimization and control of cable forces in a hybrid beam cable-stayed bridge based on a distributed algorithm. Engineering Optimization, 56(5): 720-739, 2024. https://doi.org/10.1080/0305215X.2023.2195172

[10] Shuang Wang, Bomin Huang. Distributed online optimisation in unknown dynamic environment. International Journal of Systems Science, 55(6): 1167-1176, 2024. https://doi.org/10.1080/00207721.2024.2302903

[11] Yassine Boujelben, Hasna Fourati. A distributed auction-based algorithm for virtual machine placement in multiplayer cloud gaming infrastructures. International Journal of Cloud Computing, 13(1): 80-98, 2024. https://doi.org/10.1504/IJCC.2024.136286

[12] Anitha C L, R. Sumathi. Anomaly detection in WSN IoT (Internet of Things) environment through a consensus-based anomaly detection approach. Multimedia Tools and Applications, 83(20): 58915-58934, 2023. https://doi.org/10.1007/s11042-023-17894-2

[13] Zuqing Zheng, Huaqing Li, Youcheng Niu, Enbing Su, Liping Feng. Distributed generalized Nash equilibrium seeking: A backward-reflected-forward-backward-based algorithm. Journal of the Franklin Institute, 361(1): 150-163, 2024. https://doi.org/10.1016/j.jfranklin.2023.11.033

[14] Shuang Liu, Jie Tian, Chao Zhai, Tiantian Li. Joint computation offloading and resource allocation in vehicular edge computing networks. Digital Communications and Networks, 9(6): 1399-1410, 2023. https://doi.org/10.1016/j.dcan.2022.12.002

[15] Huen Lou, Shigeru Fujimura. ADMM-Based Distributed Algorithm for Energy Management in Multi-Microgrid System. IEEJ Transactions on Electrical and Electronic Engineering, 19 (1): 79-89, 2023. https://doi.org/10.1002/tee.23953

[16] Xiaoxi Yan, Cheng Li, Kaihong Lu, Hang Xu. Random gradient-free method for online distributed optimization with strongly pseudo convex cost functions. Control Theory and Technology, 22(1): 14-24, 2023. https://doi.org/10.1007/s11768-023-00181-8

[17] Ilya Tsakunov, David Chudán. Use of Data Mining for Analysis of Czech Real Estate Market. Acta Informatica Pragensia, 12(2): 275-95, 2023. http://aip.vse.cz/doi:10.18267/j.aip.215

[18] Marcelo Vianna. Coordinating users to generate the base of the national industry –CAPRE's role in controlling imports of computers and peripherals (1976–1979). Interdisciplinary Science Reviews, 46 (4): 501-521, 2021. http://aip.vse.cz/doi:10.1080/03080188.2020.1865662

[19] Qian Guo, Chun Yang, Shaoqing Tian. Prediction of Purchase Intention among E-Commerce Platform Users Based on Big Data Analysis. Revue d'Intelligence Artificielle, 34(1): 95-100, 2020. https://doi.org/10.18280/ria.340113

[20] Pegah Malekpour Alamdari, Nima Jafari Navimipour, Mehdi Hosseinzadeh, Ali Asghar Safaei, Aso Darwesh. An image-based product recommendation for E-commerce applications using convolutional neural networks. Acta Informatica Pragensia, 11(1): 15-35, 2022. http://aip.vse.cz/doi/10.18267/j.aip.167.html